A DIGITAL COMPUTER STUDY OF THE BUCKLING OF ACTUAL IMPERFECT CYLINDERS - A MODIFICATION OF THE COMPUTER PROGRAM SATANS - THEORY AND USER'S MANUAL FOR SATANS-I AND SATANS II

Bruce Anthony Ryan

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DIGITAL COMPUTER STUDY OF THE BUCKLING
OF
ACTUAL IMPERFECT CYLINDERS -
A MODIFICATION
OF THE COMPUTER PROGRAM SATANS -
THEORY AND USER'S MANUAL
FOR SATANS-I AND SATANS-II

by

Bruce Anthony Ryan

Thesis Advisor: R. E. BALL

December 1972

A Digital Computer Study of the Buckling of Actual Imperfect Cylinders - A Modification of the Computer Program SATANS - Theory and User's Manual for SATANS-I and SATANS-II

by

Bruce Anthony Ryan
Lieutenant Commander, United States Navy
B.S., Villanova University, 1964
M.S., Naval Postgraduate School, 1972

Submitted in partial fulfillment of the requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1972

ABSTRACT

A digital computer program for the geometrically nonlinear static and dynamic response of arbitrarily loaded shells of revolution known as SATANS is modified, removing a requirement that the applied loads be symmetric about a datum meridional plane. The ability to include arbitrary naturally-occurring initial geometric imperfections of any shape, location or extent is added to the program. Two specific axially loaded circular cylinders are analyzed using actual initial imperfections. Detailed numerical studies are conducted to determine the buckling loads of the perfect and imperfect cylinders. Several sets of boundary conditions are considered. An analysis procedure for imperfect shells is proposed. The results indicate that a means to conduct a numerical analysis of an arbitrarily imperfect shell of revolution has been developed.

A study of the buckling behavior of the clamped shallow spherical cap modelled under various boundary conditions at the pole is included as an addendum.

TABLE OF CONTENTS

I. INTRODUCTION---------- 13

II. DESCRIPTION OF SATANS---------- 16

A. GEOMETRY AND COORDINATE SYSTEM---------- 16

B. METHOD OF SOLUTION---------- 17

C. USER PREPARED SUBROUTINES---------- 20

III. CONSTRUCTION OF SATANS-I---------- 23

A. PACKAGING---------- 23

B. PLOTTING ROUTINE---------- 23

C. MISCELLANEOUS ALTERATIONS---------- 25

D. USE OF SATANS-I---------- 27

IV. CONSTRUCTION OF SATANS-II---------- 28

A. INTRODUCTION OF FULL TRIGONOMETRIC SERIES EXPANSIONS---------- 28

B. INTRODUCTION OF INITIAL IMPERFECTIONS--------- 32

C. COMBINATION OF FULL TRIGONOMETRIC SERIES EXPANSIONS AND INITIAL IMPERFECTIONS---------- 36

D. SUBROUTINE MODIFICATION---------- 37

E. SATANS-II---------- 40

F. USE OF SATANS-II---------- 42

G. VALIDITY OF SATANS-II---------- 42

V. IMPERFECT SHELL ANALYSIS---------- 46

A. PREPARATION OF IMPERFECTION DATA---------- 46

1. Data Generation By Mathematical Description---------- 46

2. Data Generation by Graphical Plots-------- 51

3. Data Generation by Experimental Measurement---------- 51

4. Selection of Important Harmonics---------- 55

B. BOUNDARY CONDITIONS---------- 56

C. ANALYSIS---------- 58

1. Buckling of the Thin Cylinder---------- 58

2. Introduction of Imperfections---------- 70

3. Results---------- 79

D. DISCUSSION---------- 84

E. PROPOSED ANALYSIS PROCEDURE---------- 85

VI. CONCLUSIONS---------- 88

A. SATANS-I---------- 88

B. SATANS-II---------- 88

C. IMPERFECT ANALYSIS---------- 89

ADDENDUM

A DIGITAL COMPUTER STUDY OF THE EFFECTS OF VARIOUS ARTIFICIAL BOUNDARY CONDITIONS AT THE POLE OF A SHALLOW SPHERICAL SHELL

I. INTRODUCTION---------- 90

II. STATEMENT OF THE PROBLEM---------- 92

A. GEOMETRY AND LOADING---------- 92

B. POLE CONDITIONS---------- 94

C. TYPES OF ANALYSIS---------- 97

III. DISCUSSION---------- 98

A. THE λ=6 SHALLOW CAP---------- 98

1. Axisymmetric Analysis---------- 98

2. Bifurcation Buckling Analysis---------- 102

3. Nonlinear Nonsymmetric Collapse Analysis-- 105

B. THE λ=18 SHALLOW CAP---------- 108

C. COMPARISONS WITH OTHER RESEARCHERS---------- 108

IV. CONCLUSIONS---------- 110

APPENDIX A Nonlinear Terms---------- 112

APPENDIX B Eigenvalue Problem Matrix Element Definitions---------- 115

APPENDIX C Input Data Guide for SATANS-I and SATANS-II---------- 117

APPENDIX D Listing of SATANS-I User Prepared Subroutines---------- 124

APPENDIX E Listing of SATANS-I---------- 128

APPENDIX F Listing of Modified Subroutines and COMMON Statements for SATANS-II---------- 210

APPENDIX G Listing of Imperfection Data Handling Program---------- 252

LIST OF REFERENCES---------- 263

INITIAL DISTRIBUTION LIST---------- 265

FORM DD 1473---------- 267

LIST OF TABLES

I. Harmonic, Mode and Fourier Index Correspondence-- 38

II. Fourier Coefficients for Check Loadings---------- 44

III. Imperfect Shell Overall Characteristics---------- 52

IV. Boundary Condition Specifications---------- 59

V. Critical Buckling Loads of Individual Harmonics-- 75

VI. Relative Imperfection Rotation Coefficient Sizes, Shell A-14---------- 76

VII. Harmonic Selection, Shell A-14---------- 77

VIII. Buckling Loads, Minute Versus Full Imperfections- 80

IX. Load Steps Versus Iterations---------- 83

X. Predicted Buckling Loads of the Imperfect Shell-- 86

XI. Shallow Cap (λ=6) Buckling Loads and Displacements at a Point---------- 99

XII. Shallow Cap (λ=6) Buckling Loads Comparison------ 109

LIST OF FIGURES

1. Shell Geometry and Coordinate System---------- 18

2. Positive Directions for Variables---------- 19

3. Nonlinear Behavior Affecting Convergence---------- 21

4. Premature Program Termination---------- 26

5. SATANS-II Check Loadings---------- 43

6. Simplified Imperfection Data Generation Scheme----- 47

7. Mathematical Imperfection Generation---------- 49

8. Meridional Imperfection Rotation Representations--- 50

9. Circumferential Trace Number and Meridional Station Number Correspondence---------- 53

10. Data Point Extrapolation Technique for $\overline{W}$---------- 54

11. Cylindrical Shell Testing Configuration---------- 57

12. Membrane Prebuckle Deflection Shape, Shell A-14---- 62

13. Critical Buckling Modes from Eigenvalue Analysis, Shell A-14---------- 64

14. Buckling Load vs Stations, Membrane Prebuckle Analysis, Shell A-14---------- 66

15. Buckling Load vs Stations, Consistent Analysis, Shell A-14---------- 67

16. Buckling Load vs Buckling Mode, Membrane Prebuckle Analysis, Shell A-14---------- 68

17. Buckling Load vs Buckling Mode, Consistent Analysis, Shell A-14---------- 69

18. Circumferential Imperfection Rotation Representation, Φ_θ vs θ, Plate 1---------- 71
Plate 2---------- 72

19. Critical Buckling Load vs Imperfection Level------- 82

20. λ=6 Shallow Spherical Cap Geometry and Loading----- 93

21. Normal Displacement Error, Axisymmetric Analysis, λ=6 Cap---------- 100

22. Meridional Force Error, Axisymmetric Analysis, λ=6 Cap---------- 101

23. Normal Displacement Error, Bifurcation Buckling Analysis, λ=6 Cap---------- 103

24. Meridional Force Error, Bifurcation Buckling Analysis, λ=6 Cap---------- 104

25. Normal Displacement Error, Nonsymmetric Collapse Analysis, λ=6 Cap---------- 106

26. Meridional Force Error, Nonsymmetric Collapse Analysis, λ=6 Cap---------- 107

TABLE OF SYMBOLS

a	= reference length
b	= nondimensional inplane stiffness
E	= Young's modulus
E_o	= reference Young's modulus
h	= thickness
h_o	= reference thickness
H	= rise of a shallow spherical cap
m	= harmonic number, meridional
M_s, M_θ, $M_{s\theta}$	= bending and twisting moments per unit length
m_s, m_θ, $m_{s\theta}$	= nondimensional bending and twisting moment series coefficients
n	= harmonic number, circumferential
N_s, N_θ, $N_{s\theta}$	= membrane forces per unit length
p	= nondimensional pressure load series coefficient
$\hat{Q}_s$	= effective transverse force
q, q_s, q_θ	= normal, meridional and circumferential components of applied pressure loading
q/q_o	= ratio of applied load to critical load
R_s, R_θ	= principal radii of curvature
r	= normal distance from the axis of the shell
s	= meridional shell coordinate
t_s, t_θ, $t_{s\theta}$	= nondimensional membrane force series
t_T	= nondimensional thermal membrane force series coefficient
U, V	= displacements tangent to the meridian and tangent to the parallel circle

u, v = nondimensional series coefficients of U, V

W = displacement normal to the reference surface

$\overline{W}$ = displacement normal to the reference surface due to initial imperfections

w = nondimensional series coefficient of W

β, β_s, β_θ, $\beta_{s\theta}$ = nondimensional nonlinear term series coefficients in the strain-displacement equations

Δ = distance between stations

ε_s, ε_θ, $\varepsilon_{s\theta}$ = nondimensional reference surface strain coefficients

$\overline{\varepsilon}_s$, $\overline{\varepsilon}_\theta$, $\overline{\varepsilon}_{s\theta}$ = nondimensional reference surface imperfection strain series coefficients

ζ = shell coordinate perpendicular to the reference surface

θ = shell circumferential coordinate

η, η_s, η_θ, $\eta_{s\theta}$ = nondimensional nonlinear term series coefficients in the equilibrium equations

λ = nondimensional geometric descriptive parameter for a spherical cap

ν = Poisson's ratio

σ_o = reference stress level

ρ = nondimensional radius

ω_s, ω_θ = nondimensional curvatures

Φ, Φ_s, Φ_θ = reference surface rotations

$\overline{\Phi}_s$, $\overline{\Phi}_\theta$ = reference surface rotations due to initial imperfections

ϕ, ϕ_s, ϕ_θ = nondimensional series coefficients for reference surface rotations

$\overline{\phi}_s$, $\overline{\phi}_\theta$ = nondimensional series coefficients for reference surface rotations due to initial imperfections

$[\Omega]$, $[\Lambda]$ = 4 x 4 boundary condition matrices

$\{\ell\}$ = 1 x 4 boundary condition matrix

$\underline{[E]}$, $\underline{[F]}$, $\underline{[G]}$ = matrices [Refs. 3 and 4]

$\sum_{1}^{\infty}$ = a summation, on the harmonic index number n, where n = 1, 2, $\cdots$, ∞

SUPERSCRIPTS

^{1} = indicates the coefficient of a previously existing term

^{2} = indicates the coefficient of a newly introduced term

^{12} = indicates the coefficient of the product of a previously existing term and a newly introduced term

(n) = Fourier index

$'$ = (prime), partial differentiation in the meridional direction

$\cdot$ = (dot), partial differentiation in the circumferential direction

ACKNOWLEDGEMENT

The author wishes to extend his finest appreciation to Dr. Robert E. Ball for his continuous and highly professional guidance and trust with "his program." The insight, foresight and encouragement provided by Dr. Ball were major factors leading to the timely completion of this effort and in increasing its worth. Appreciation is also extended to Drs. Johann Arbocz and Charles Babcock at the California Institute of Technology for the use of their experimental data.

The author also wishes to thank Drs. Richard F. Hartung, Bö O. Almroth, Frank A. Brogan and David Bushnell, and the Lockheed Missiles and Space Company's Palo Alto Research Laboratory , Structural Analysis Division for the personal education they imparted, for their aid in the preparation of the addendum to this thesis, and for the use of their structural analysis programs.

I. INTRODUCTION

Shells are used as structural elements in nearly every space vehicle, aircraft or other structure where light weight is of importance. Consequently, a large number of digital computer programs for the structural analysis of this shells have been developed. Many of these programs were briefly described by Bushnell in Ref. 1. An assessment concerning the analysis capability based on computational systems that have application to a broad class of shell structures was presented by Hartung in Ref. 2. In his assessment, Hartung discusses the limitations, scope, accuracy and efficiency of virtually every major shell structural analysis program currently available and under development. Of particular interest here, is the computer program SATANS -- Static And Transient Analysis, Nonlinear, Shells -- developed by Ball for the geometrically nonlinear analysis of arbitrarily loaded shells of revolution /Refs. 3 and 4/.

First, in section II, a brief description of SATANS is presented including the basic theory, the program capabilities and limitations, the solution method and procedure and some general comments.

Second, in section III, SATANS-I is constructed. Modifications to SATANS making the program more user oriented are discussed. Included in these modifications are a reorganization of most of SATANS into a 'storable package,' incorporation of an equipment-independent plotting package

which uses a high-speed line printer, modification of iteration control, discussion of extrapolation technique and alteration of boundary load application procedures.

Third, and most important, the subject of imperfections is treated in section IV. Thin isotropic shells of revolution have consistently displayed experimental buckling loads much lower than the theoretically predicted buckling loads. This disagreement has generally been attributed to (1), the mathematical inability to correctly model the boundary conditions as they exist on the test specimen, and (2), a high degree of imperfection sensitivity, i.e., naturally-occurring imperfections in the initial shape of the shell cause a premature triggering of critical buckling modes and early shell collapse. An attempt by Dantone [Ref. 5] to modify SATANS and to include initial imperfections in the analysis of cylindrical shells was severely hampered by a restriction on the use of SATANS, namely, all applied loading and initial conditions -- and hence, initial imperfections -- must be symmetric about some meridional plane. Discussion of the removal of this restriction and introduction of initial imperfections into the program are covered in this section.

Fourth, and also in section IV, the implementation of the initial imperfection capability is presented. The resulting program is called SATANS-II. Steps for actual construction of SATANS-II from SATANS-I are presented as are instructions for its use.

Fifth, in section V, actual imperfect shell analyses are performed on two specific axially loaded circular cylinders. The actual topography of the cylinders has been measured by Arbocz and Babcock [Ref. 6] and this data provides the opportunity to include actual naturally-occurring initial imperfections into the analysis by SATANS-II. Detailed numerical studies are performed to determine the classical and consistent bifurcation buckling loads of the perfect cylinders and the results are compared with previously published results. Stability solutions for the imperfect cylinders under six boundary conditions are obtained and compared with experimental values. An analysis procedure for imperfect shells is proposed based upon these results.

Lastly, as an addendum to this report, the clamped shallow spherical cap is investigated under various boundary conditions at the pole. While on an industrial tour with the Structural Analysis Division of the Palo Alto Research Laboratory of the Lockheed Missiles and Space Company, Sunnyvale, California, Ryan noted that few shell structural analysis programs have the capability of analyzing a shell of revolution which has an initial or a final pole -- continuous structure through the axis of revolution. This investigation attempted to determine the effects on the solution caused by substitution of a small hole or a small rigid plug for the pole.

II. DESCRIPTION OF SATANS

The computer program known as SATANS is capable of performing static and transient analyses of thin, linearly-elastic shells of revolution subjected to arbitrary loads and temperature distributions. It is based on Sanders' nonlinear field equations [Ref. 7] for the conditions of small strains and moderately small rotations. The program can be used under the following conditions:

1. The geometric and material properties of the shell must be axisymmetric, but may vary along the shell meridian.

2. The shell material must be isotropic, but the modulus of elasticity may vary through the thickness.

3. Poisson's ratio must be constant.

4. The boundaries of the shell may be closed, fixed, elastically restrained or free, and loaded or unloaded.

5. The applied forces, pressures, temperatures and initial conditions must be symmetric about a datum meridional plane.

A. GEOMETRY AND COORDINATE SYSTEM

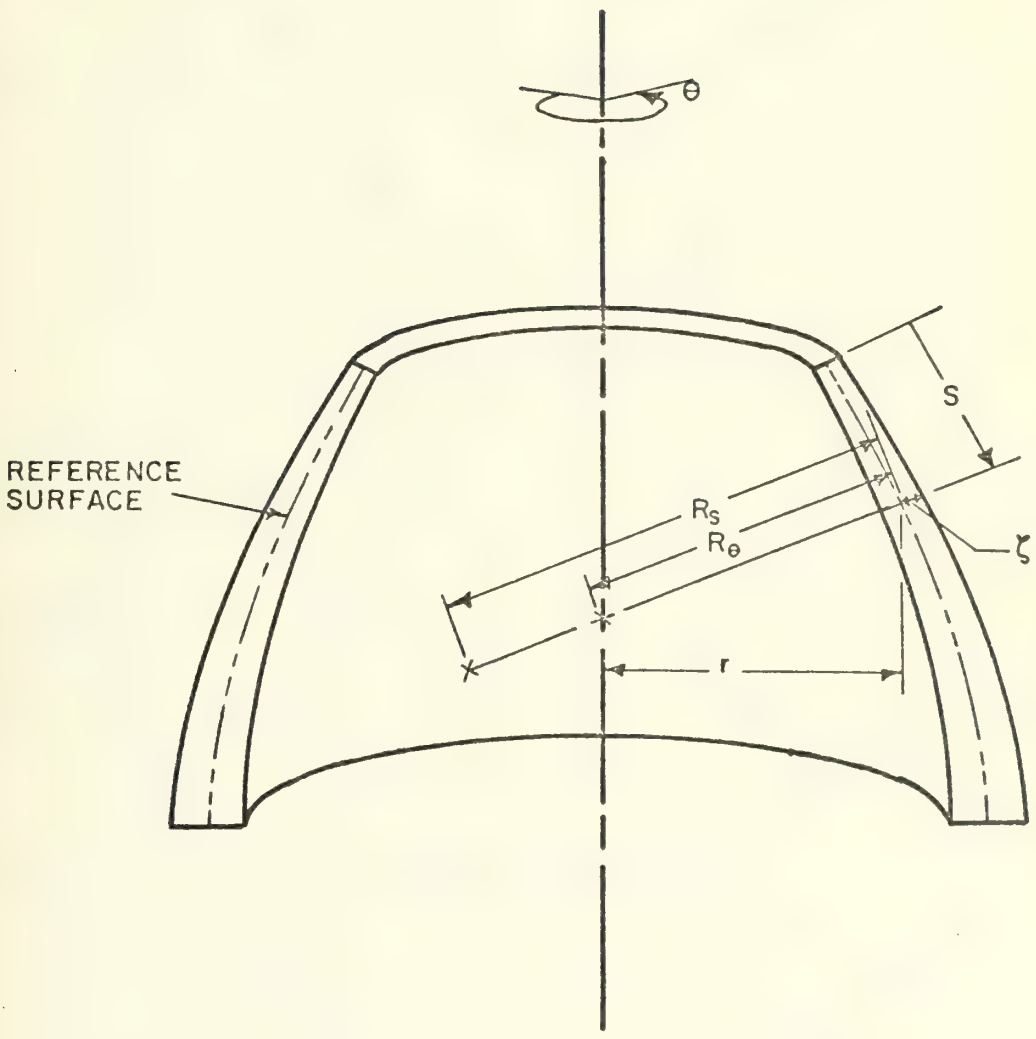
The coordinate system used to locate points within the shell is the s, θ, ζ orthogonal system, where s is the meridional distance along a reference surface, θ is the circumferential angle measured about the axis of revolution and ζ is the perpendicular distance from the reference surface. The shell geometry and coordinate system are shown

in Figure 1. Positive directions for displacements, rotations, forces, moments and loads are shown in Figure 2.

B. METHOD OF SOLUTION

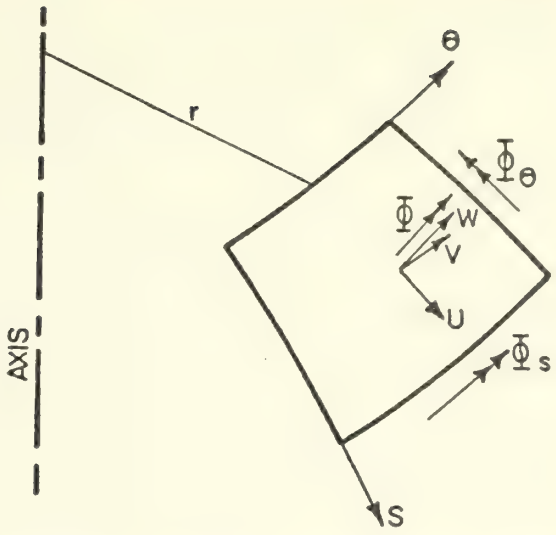
The governing partial differential equations are reduced to an infinite number of sets of four second order differential equations in the meridional and time coordinates by expanding all dependent variables in a sine or cosine series in terms of $n\theta$, where n is the Fourier index (harmonic) and θ is the circumferential coordinate. The sets are uncoupled by utilizing appropriate trigonometric identities and by treating the nonlinear coupling terms as 'pseudo loads.' The meridional derivatives are replaced by conventional central finite difference approximations, and the displacement accelerations are approximated by the implicit Houbolt backward differencing scheme [Ref. 8]. This leads to sets of algebraic equations in terms of the dependent variables and the Fourier index or harmonic. At each load or time step, an estimate of the solution is obtained by extrapolation from the solutions at previous load or time steps. The sets of algebraic equations are repeatedly solved using Potters' form of Gaussian elimination [Ref. 9]. The pseudo loads are recomputed after each solution and the procedure continues until the solution converges.

In the static analysis, a load-displacement history is obtained by an automatic incrementing of applied loadings in equal steps until the number of iterations required to achieve convergence in any harmonic exceeds a prescribed

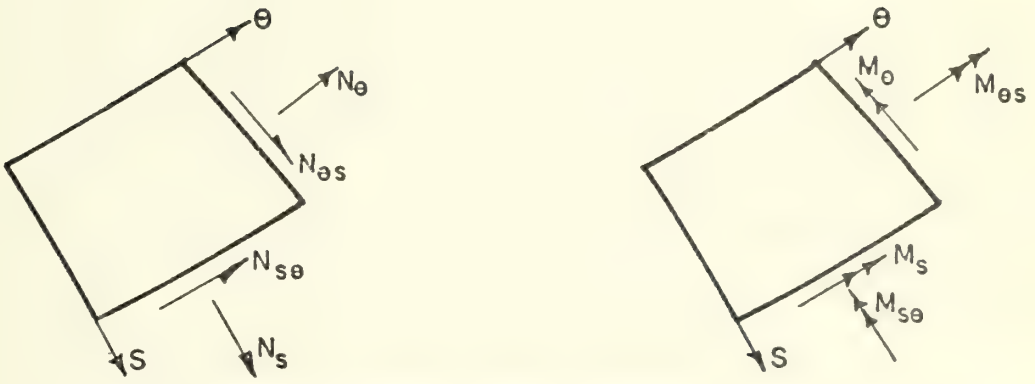


SHELL GEOMETRY AND COORDINATE SYSTEM

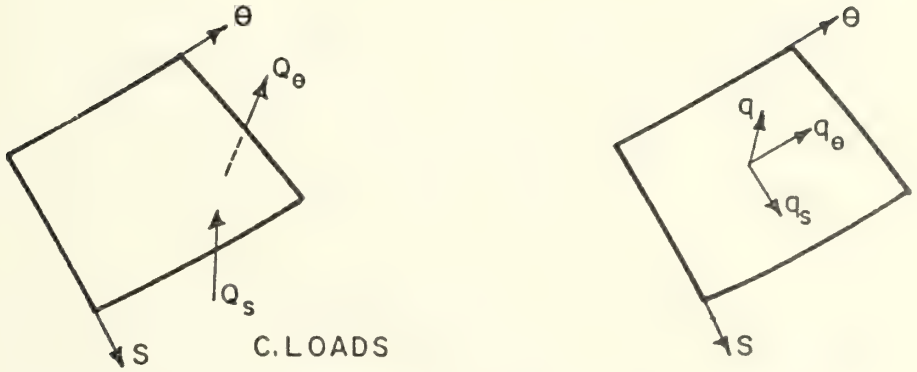
FIGURE I



A. DISPLACEMENTS AND ROTATIONS



B. FORCES AND MOMENTS



C. LOADS

POSITIVE DIRECTIONS FOR VARIABLES

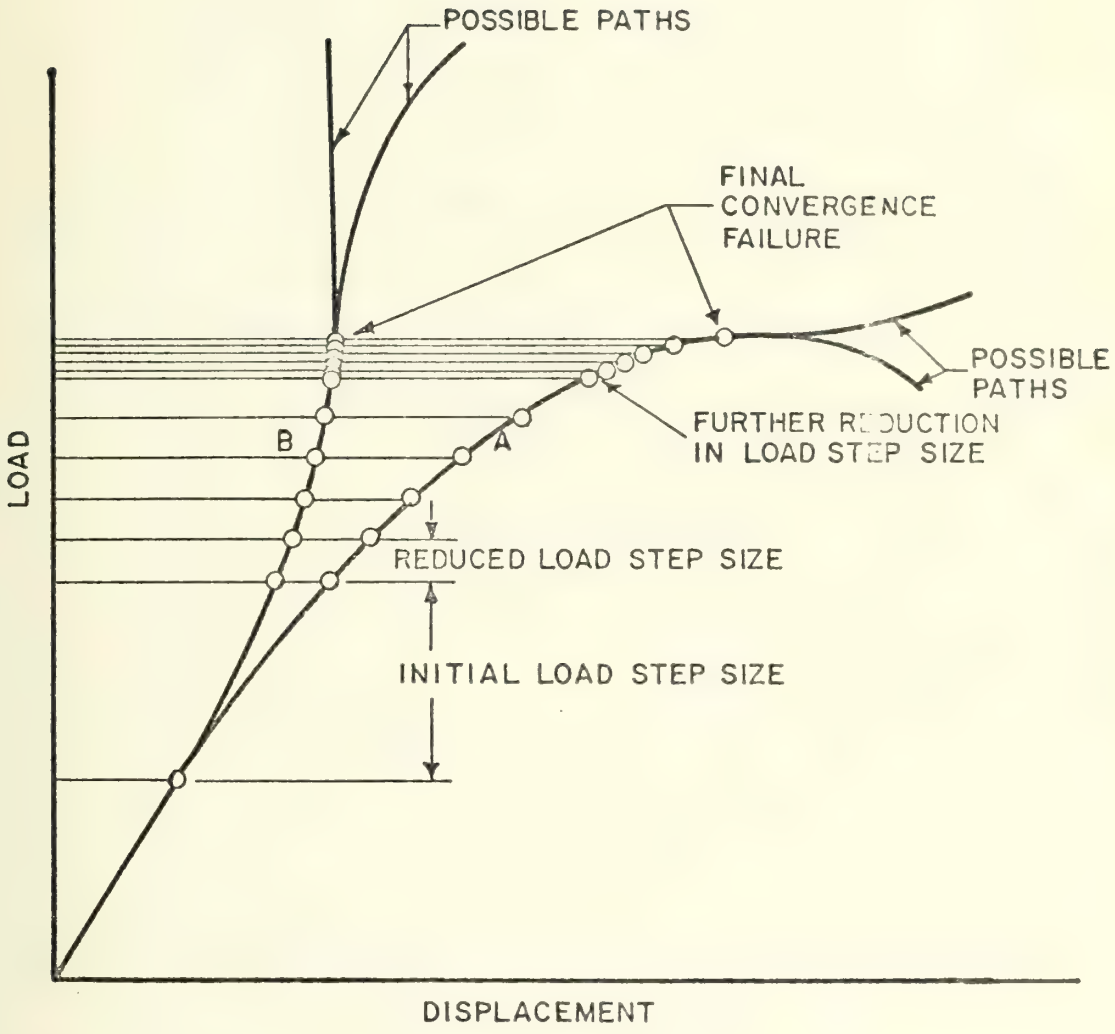
FIGURE 2

limit. The load increment is then reduced by a factor of five and the procedure continues until a prescribed number of load steps, or a prescribed number of load step reductions have been made.

Since the method of solution is based on the pseudo load approach, the shell reacts equally, in a linear fashion, to any change in either the applied load or the pseudo load. Thus, failure of the solution to converge in any harmonic can be attributed to two types of nonlinear behavior, both of which are illustrated in Figure 3. Referring to the figure, curve A experiences an inflection point or a maximum. Either condition causes a zero or nearly zero slope and a convergence failure. This curve is typical of softening nonlinearity. Curve B, on the other hand, is typical of a stiffening nonlinearity and a convergence failure will occur when the slope becomes large. It is necessary, then, to examine the load-displacement behavior of the shell to determine the cause of convergence failure. A detailed discussion of the theory and implementation of SATANS can be found on pages 8 through 29 and in Appendices B, C and D of Ref. 3, and on pages 8 through 28, pages 56 through 61 and page 100 of Ref. 4.

C. USER PREPARED SUBROUTINES

Five small subroutines describing the shell geometry, the stiffnesses, the loading (physical and/or thermal), and the initial conditions (in the dynamic case) must be prepared by the user. Complete descriptions for the



NONLINEAR BEHAVIOR AFFECTING CONVERGENCE

FIGURE 3

preparation of these subroutines can be found on pages 34 through 38 of Ref. 4.

III. CONSTRUCTION OF SATANS-I

Stein [Ref. 10] observes that a basic impediment to the widespread use of existing computer programs is the difficult and laborious process of assimilation of a program's basic theory, its implementation and its use by a prospective researcher. He also warns the 'unwary analyst' of the dangers inherent in computer output, which is a 'blizzard of numbers.' As a consequence, SATANS is continually being modified in an attempt to make it more usable. The most recent modifications made for this purpose are described in this section. This new version of SATANS is called SATANS-I.

A. PACKAGING

SATANS has been reorganized to allow 99% of the code to be prepackaged for repeated use by researchers. Compiled versions of all subroutines and main controlling routines can now be placed on a disk, in a data cell or on a magnetic tape, allowing the user to 'submit' only the user written subroutines and input data. A complete listing of this storable package is given in Appendix E.

B. PLOTTING ROUTINE

A great advantage is gained if the computer output can be presented in graphical format. Such plots, however, are often generated using mechanical or photo-electronic devices which are not universally usable. Consequently, SATANS has

been enhanced with an equipment-independent plot package utilizing self-contained subroutines and the high-speed line printer.

The main program listed in Ref. 4 could not be modified to incorporate the logic and control for a graphical ability as it then exceeded the capability of the FORTRAN-H compiler. Therefore, the main program of Ref. 4 has been replaced by a new main program and two subroutines described as follows:

SATANS - This is the new 'main.' It reads all data cards, prepares the title page of output and passes control to either of two basic controlling subroutines.

STATIC - This is the controlling subroutine for a static analysis. It conducts initialization, establishes geometry, stiffness, loading and boundary conditions, controls plotting of input information and controls the solution procedure.

DYNAMC - This is the controlling subroutine for a dynamic analysis and it operates in a fashion similar to subroutine STATIC.

The graphical ability is obtained by the introduction of two controlling subroutines, PLOT1 and PLOT2, four plotting subroutines, PLOTIT, SCALIT, ROUND and DRAWIT,[1] and modification of subroutine OUTPUT for the selection of output data for plotting. A single data card with request information

---

[1]The four plotting subroutines are slightly modified versions of those contained in Ref. 11 under the listing "JS UTPLOT."

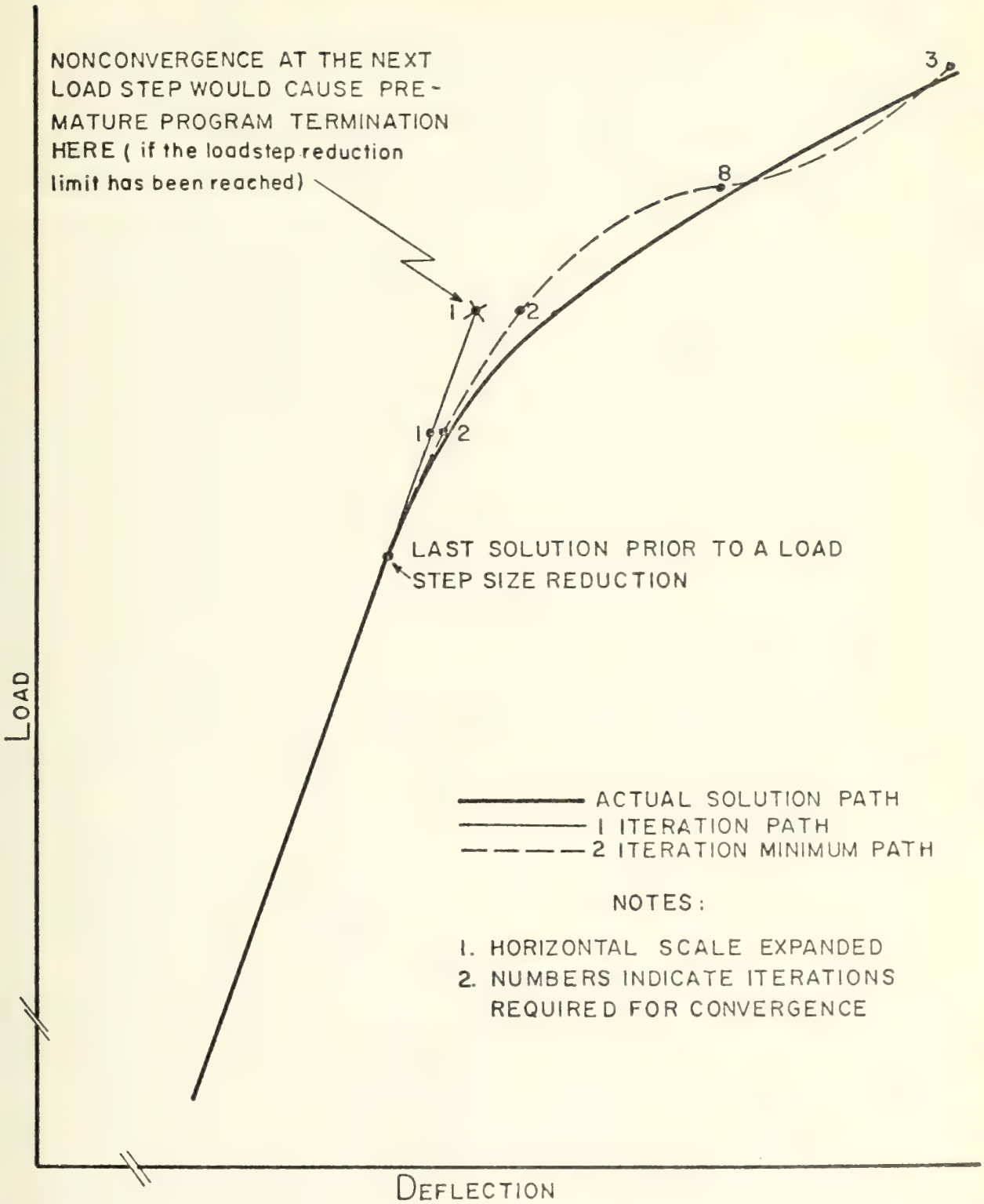
is required to generate the plots. The plotting routines will handle a maximum of 100 stations. This routine is included in the listing given in Appendix E.

C. MISCELLANEOUS ALTERATIONS

The iteration lower limit has been raised to two iterations. This change was introduced because it was noted that immediately after a load step size reduction, several consecutive solutions were obtained in one iteration, a completely linear process. This was then followed by non-convergence and, in some instances, premature program termination. Setting a minimum of two iterations automatically introduces nonlinearity into each load step and appears to solve this problem, as illustrated in Figure 4.

A quadratic extrapolation technique was investigated using a static analysis of the highly nonlinear shallow spherical cap. No gain in solution accuracy was seen and a price was paid in program execution time due to approximately 10% greater total iterations required, not to mention the increased storage space requirement. This phenomenon is exactly as observed by Stricklin et al [Ref. 12], hence the linear extrapolation technique has been retained.

Since boundary loads are introduced through the boundary conditions and since the program considers the boundary conditions to be the same in every harmonic, changes indicated in Appendix B of Ref. 5 are permanently incorporated into the program. This alteration causes the applied



PREMATURE PROGRAM TERMINATION

FIGURE 4

loading at a boundary to be taken in the zeroeth harmonic only, i.e., when n=0.

All 'COMMON' statements have been ordered and separated and some slightly modified. This alteration allows easier reading and correlation of FORTRAN variables with specific subroutines.

The output ability has been expanded to allow printed data (and plots) at up to 36 vice six meridians.

In the case of a shell with an initial (or a final) pole, determination of the shear force at the pole in the n=1 harmonic has been deemed an inconsequential and lengthy computation and has been removed. Appropriate changes have been made in subroutines INLPOL, FNLPOL and OUTPUT.

D. USE OF SATANS-I

Instructions for preparation of input data cards for SATANS-I are contained in Appendix C and the initiating main program and user written subroutines appear in Appendix D. Appendices D and E contain a complete listing of the computer program, SATANS-I.

SATANS-I requires 206 k-bytes of main core when run on an IBM-360/67 computer under OS/MVT and compiled by the FORTRAN-H compiler at optimization level 2. Execution times are the same as for SATANS unless plotting is requested, and then is increased approximately 0.01 seconds for each 100 point plot generated.

IV. CONSTRUCTION OF SATANS-II

As discussed previously, a serious impediment to the consideration of initial imperfections by Dantone [Ref. 5] was the requirement that the imperfections be symmetric about a datum meridian. This restriction results from the fact that all dependent variables were expanded in a single sine or cosine series. The analysis presented in this section removes this restriction by introducing the use of full trigonometric series expansions for the dependent variables, and then modifies the resulting new relations where required for the introduction of initial imperfections. The computer program that incorporates these changes is a modification to SATANS-I, called SATANS-II.

A. INTRODUCTION OF FULL TRIGONOMETRIC SERIES EXPANSIONS

The original assumption by Ball [Ref. 3] was that a normal pressure load could be expressed in a Fourier cosine series as

$$q = (\sigma_o h_o / E_o) \sum_{o}^{\infty} p^{(n)} \text{Cos } n\theta \qquad (1)$$

where σ_o = the reference stress level,

h_o = the reference thickness,

E_o = the reference Young's modulus,

n = the Fourier index, or harmonic,

and p = the nondimensional Fourier coefficient of the normal pressure load. The normal displacement W

corresponding to this pressure was taken as

$$W = (a\sigma_o/E_o) \sum_{0}^{\infty} w^{(n)} \text{Cos } n\theta \tag{2}$$

where a is the shell characteristic length and w is the nondimensional series coefficient of the normal displacement. For a completely arbitrary case, representing q and W with the complete Fourier series gives

$$q = (\sigma_o h_o/E_o) \left(\sum_{0}^{\infty} p^{1}(n) \text{Cos } n\theta + \sum_{1}^{\infty} p^{2}(n) \text{Sin } n\theta \right) \tag{3}$$

and

$$W = (a\sigma_o/E_o) \left(\sum_{0}^{\infty} w^{1}(n) \text{Cos } n\theta + \sum_{1}^{\infty} w^{2}(n) \text{Sin } n\theta \right) \tag{4}$$

where superscript ^{1} indicates a previously existing coefficient and superscript ^{2} indicates a new coefficient. Similar series expansions can be written for the other pressures, displacements, etc.

The nonlinear terms associated with the full series are illustrated by Φ_s^2,

$$\Phi_s^2 = (\sigma_o/E_o)^2 \left\{ \left(\sum_{0}^{\infty} \phi_s^{1}(n) \text{Cos } n\theta \right)^2 + \left(\sum_{1}^{\infty} \phi_s^{2}(n) \text{Sin } n\theta \right)^2 \right.$$
$$\left. + 2 \left(\sum_{0}^{\infty} \phi_s^{1}(n) \text{Cos } n\theta \right) \left(\sum_{1}^{\infty} \phi_s^{2}(n) \text{Sin } n\theta \right) \right\} \tag{5}$$

where the ϕ_s are the nondimensional series coefficients of

the meridional rotation, given as

$$\phi_s^{1(n)} = \left(-w^{1(n)}\right)' + \omega_s u^{1(n)}$$

$$\phi_s^{2(n)} = \left(-w^{2(n)}\right)' + \omega_s u^{2(n)} \tag{6}$$

Originally, in Refs. 3 and 4, the nonlinear term Φ_s^2 was represented as

$$\Phi_s^2 = \left(\sigma_o/E_o\right)^2 \left(\sum_0^\infty \phi_s^{1(n)} \text{Cos } n\theta \right)^2 \equiv \sigma_o/E_o \sum_0^\infty \beta_s^{1(n)} \text{Cos } n\theta \tag{7}$$

since

$$\text{Cos}(n\theta)\text{Cos}(m\theta) = \tfrac{1}{2}\text{Cos}(n-m)\theta + \tfrac{1}{2}\text{Cos}(n+m)\theta$$

The $\beta_s^{1(n)}$ term contains the products of all the ϕ_s's whose harmonic sum or difference equals n. There are now two additional terms in equation (5). These can be represented as

$$\left(\sigma_o/E_o\right)^2 \left(\sum_1^\infty \phi_s^{2(n)} \text{Sin } n\theta \right)^2 \equiv \sigma_o/E_o \sum_1^\infty \beta_s^{2(n)} \text{Cos } n\theta \tag{8}$$

and

$$2\left(\sigma_o/E_o\right)^2 \left(\sum_0^\infty \phi_s^{1(n)} \text{Cos } n\theta \right) \left(\sum_1^\infty \phi_s^{2(n)} \text{Sin } n\theta \right)$$

$$\equiv \sigma_o/E_o \sum_1^\infty \beta_s^{12(n)} \text{Sin } n\theta \tag{9}$$

where superscript ^{12} indicates the product of a previously existing coefficient and a new coefficient.

Thus,

$$\Phi_s^2 = \sigma_o/E_o \left(\sum_0^\infty \beta_s^1(n) \operatorname{Cos} n\theta + \sum_0^\infty \beta_s^2(n) \operatorname{Cos} n\theta + \sum_1^\infty \beta_s^{12}(n) \operatorname{Sin} n\theta \right) \tag{10}$$

Similarly, the nonlinear term Φ_θ^2 (and also Φ^2 since it is of the same form) is given by

$$\Phi_\theta^2 = (\sigma_o/E_o)^2 \left[\left(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \right)^2 + \left(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \right)^2 + 2 \left(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \right) \left(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \right) \right] \tag{11}$$

which can be represented as

$$\Phi_\theta^2 = \sigma_o/E_o \left(\sum_0^\infty \beta_\theta^1(n) \operatorname{Cos} n\theta + \sum_0^\infty \beta_\theta^2(n) \operatorname{Cos} n\theta + \sum_1^\infty \beta_\theta^{12}(n) \operatorname{Sin} n\theta \right) \tag{12}$$

The nonlinear term $\Phi_s \Phi_\theta$ is given by

$$\begin{aligned} \Phi_s \Phi_\theta = (\sigma_o/E_o)^2 \Bigg[& \left(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \right) \left(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \right) \\ & + \left(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \right) \left(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \right) \\ & + \left(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \right) \left(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \right) \\ & + \left(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \right) \left(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \right) \Bigg] \end{aligned} \tag{13}$$

which can be represented as

$$\Phi_s\Phi_\theta = \sigma_o/E_o \left(\sum_1^\infty \beta_{s\theta}^{1}(n) \operatorname{Sin} n\theta + \sum_0^\infty \beta_{s\theta}^{12}(n) \operatorname{Cos} n\theta + \sum_0^\infty \beta_{s\theta}^{2}(n) \operatorname{Sin} n\theta \right) \tag{14}$$

Following this scheme, similar expressions can be obtained for the remaining nonlinear terms $\Phi_s N_s$, $\Phi_\theta N_\theta$, $\Phi_s N_{s\theta}$, $\Phi_\theta N_{\theta s}$, ΦN_s and ΦN_θ.

All of the nonlinear terms fall into one of the three categories $\operatorname{Cos}(k\theta)\operatorname{Cos}(\ell\theta)$, $\operatorname{Sin}(k\theta)\operatorname{Sin}(\ell\theta)$ or $\operatorname{Cos}(k\theta)\operatorname{Sin}(\ell\theta)$. Since these three forms were already accounted for in SATANS, there was no need to develop any new routines for the full series expansions. It was only necessary to introduce each new nonlinear term into the appropriate existing summation in subroutines PHIBET and TEAETA.

B. INTRODUCTION OF INITIAL IMPERFECTIONS

The initial imperfections only influence the nonlinear terms in the equations. The strain-displacement relationships used in this analysis are

$$\begin{aligned} \varepsilon_s &= U' + W/R_s + \tfrac{1}{2}\left(\Phi_s^2 + \Phi^2\right) \\ \varepsilon_\theta &= V^\cdot/r + r'U/r + W/R_\theta + \tfrac{1}{2}\left(\Phi_\theta^2 + \Phi^2\right) \\ \varepsilon_{s\theta} &= \tfrac{1}{2}\left(V' + U^\cdot/r - r'V/r + \Phi_s\Phi_\theta\right) \end{aligned} \tag{15}$$

where

$$\Phi_s = -W' + U/R_s$$

$$\Phi_\theta = -W^{\cdot}/r + V/R_\theta \tag{16}$$

$$\Phi = \tfrac{1}{2}(V' + r'V/r - U^{\cdot}/r)$$

and where ε_s, ε_θ, $\varepsilon_{s\theta}$ represent the reference surface strains, Φ_s, Φ_θ, Φ represent the reference surface rotations, and superscript primes and dots represent partial differentiation with respect to s and θ, respectively.

Consider a shell whose geometry differs slightly from a shell of revolution. For this imperfect shell, a deviation from a specific shape occurs in the form of a transverse displacement, $\bar{W}$. This $\bar{W}$ produces no strain. Defining W* as a total displacement,

$$W^* \equiv W + \bar{W} \tag{17}$$

and replacing W in equations (15) with W* of equation (17) gives

$$\varepsilon_s + \bar{\varepsilon}_s = U' + W^*/R_s + \tfrac{1}{2}(\Phi_s^{*2} + \Phi^2)$$

$$\varepsilon_\theta + \bar{\varepsilon}_\theta = V^{\cdot}/r + r'U/r + W^*/R_\theta + \tfrac{1}{2}(\Phi_\theta^{*2} + \Phi^2) \tag{18}$$

$$\varepsilon_{s\theta} + \bar{\varepsilon}_{s\theta} = \tfrac{1}{2}(V' + U^{\cdot}/r - r'V/r + \Phi_s^*\Phi_\theta^*)$$

where

$$\Phi_s^* = -W^{*\prime} + U/R_s$$

$$\Phi_\theta^* = -W^{*\cdot}/r + V/R_\theta \tag{19}$$

and $\bar{\varepsilon}_s$, $\bar{\varepsilon}_\theta$ and $\bar{\varepsilon}_{s\theta}$ are the pseudo strains due to $\bar{W}$ only. When $U = V = W = 0$, then $W^* = \bar{W}$, and $\varepsilon_s \equiv \varepsilon_\theta \equiv \varepsilon_{s\theta} \equiv 0$. Substituting these relationships into equation (18), solving for $\bar{\varepsilon}_s$, $\bar{\varepsilon}_\theta$, $\bar{\varepsilon}_{s\theta}$ and placing these terms in equations (16) lead to

$$\begin{aligned}
\varepsilon_s &= U' + W^*/R_s + \tfrac{1}{2}\left(\Phi_s^{*2} + \Phi^2\right) - \bar{W}/R_s - \tfrac{1}{2}(\bar{W}')^2 \\
\varepsilon_\theta &= V^\cdot/r + r'U/r + W^*/R_\theta + \tfrac{1}{2}\left(\Phi_\theta^{*2} + \Phi^2\right) - \bar{W}/R_\theta - \tfrac{1}{2}(\bar{W}^\cdot)^2/r \\
\varepsilon_{s\theta} &= \tfrac{1}{2}\left(V' + U^\cdot/r - r'V/r + \Phi_s^*\Phi_\theta^* - \bar{W}'\bar{W}^\cdot/r\right)
\end{aligned} \tag{20}$$

Expansion of equations (20) making use of equations (16), (17) and (19) lead to

$$\begin{aligned}
\varepsilon_s &= U' + W/R_s + \tfrac{1}{2}\left(\Phi_s^2 + \Phi^2\right) - \{\bar{W}'\Phi_s\} \\
\varepsilon_\theta &= V^\cdot/r + r'U/r + W/R_\theta + \tfrac{1}{2}\left(\Phi_\theta^2 + \Phi^2\right) - \{\bar{W}^\cdot\Phi_\theta/r\} \\
\varepsilon_{s\theta} &= \tfrac{1}{2}\left(V' + U^\cdot/r - r'V/r + \Phi_s\Phi_\theta - \{\bar{W}'\Phi_\theta + \bar{W}^\cdot\Phi_s/r\}\right)
\end{aligned} \tag{21}$$

where the terms enclosed in braces are the newly generated terms due to the imperfections, and

$$\begin{aligned}
\Phi_s^* &= \Phi_s - \{\bar{W}'\} \\
\Phi_\theta^* &= \Phi_\theta - \{\bar{W}^\cdot/r\}
\end{aligned} \tag{22}$$

The presence of imperfections has no effect on the changes of curvatures since the curvature-displacement relations are linear.

Expressing the imperfection partial derivatives $-\bar{W}'$ and $-\bar{W}^{\cdot}/r$ as imperfection rotations $\bar{\Phi}_s$ and $\bar{\Phi}_\theta$, the strain-displacement equations (21) can be expressed as

$$\varepsilon_s = U' + W/R_s + \tfrac{1}{2}\left(\Phi_s^2 + \{2\Phi_s\bar{\Phi}_s\} + \Phi^2\right)$$

$$\varepsilon_\theta = V^{\cdot}/r + r'U/r + W/R_\theta + \tfrac{1}{2}\left(\Phi_\theta^2 + \{2\Phi_\theta\bar{\Phi}_\theta\} + \Phi^2\right) \quad (23)$$

$$\varepsilon_{s\theta} = \tfrac{1}{2}\left(V' + U^{\cdot}/r - \Phi_s\Phi_\theta + \{\Phi_\theta\bar{\Phi}_s + \Phi_s\bar{\Phi}_\theta\}\right)$$

where the terms in braces are the newly generated terms. Imperfection rotations also affect the nonlinear term, $\Phi_s N_s$ (and the others listed after equation (14)). For these terms, the reference surface rotations Φ_s and Φ_θ must be replaced by Φ_s^* and Φ_θ^* of equations (22).

To illustrate the procedure which is used in the following section to introduce the initial imperfection rotations, consider the single series expansion, equation (7), repeated here,

$$\Phi_s^2 = \left(\sigma_o/E_o\right)^2\left(\sum_0^\infty \phi_s^{1}(n)\,\mathrm{Cos}\ n\theta\right)^2 \equiv \sigma_o/E_o \sum_0^\infty \beta_s^{1}(n)\,\mathrm{Cos}\ n\theta \quad (7)$$

Modifying this equation to include the effect of initial imperfections indicated by equations (23) results in

$$\Phi_s^2 + \{2\Phi_s\bar{\Phi}_s\} = \left(\sigma_o/E_o\right)^2\left\{\left(\sum_0^\infty \phi_s^{1}(n)\,\mathrm{Cos}\ n\theta\right)^2 + 2\left(\sum_0^\infty \phi_s^{1}(n)\,\mathrm{Cos}\ n\theta\right)\left(\sum_0^\infty \bar{\phi}_s^{1}(n)\,\mathrm{Cos}\ n\theta\right)\right\} \quad (24)$$

$$\equiv \sigma_o/E_o \sum_0^\infty \beta_s^{*1}(n)\,\mathrm{Cos}\ n\theta$$

A redefinition of β_s to include the cross-product term is now required, along with redefinitions of the other β terms and the η terms in the other nonlinear term expressions. A redefinition of these terms alone is sufficient for the introduction of initial imperfections into a SATANS analysis where only the single series expansions are used. This was the thrust of the thesis by Dantone.

C. COMBINATION OF FULL TRIGONOMETRIC SERIES EXPANSIONS AND INITIAL IMPERFECTIONS

Consider that for every meridional rotation Φ_s, where

$$\Phi_s = (\sigma_o/E_o)^2 \left(\sum_0^\infty \phi_s^1(n) \text{Cos } n\theta + \sum_1^\infty \phi_s^2(n) \text{Sin } n\theta \right) \qquad (25)$$

there exists an initial meridional imperfection rotation $\overline{\Phi}_s$, defined as

$$\overline{\Phi}_s = (\sigma_o/E_o)^2 \left(\sum_0^\infty \overline{\phi}_s^1(n) \text{Cos } n\theta + \sum_1^\infty \overline{\phi}_s^2(n) \text{Sin } n\theta \right) \qquad (26)$$

The nonlinear term Φ_s^2, previously given by equation (5), now becomes $(\Phi_s^2 + 2\Phi_s\overline{\Phi}_s)$, according to equation (23), where Φ_s^2 is given by equation (5) and $\Phi_s\overline{\Phi}_s$ is the product of equations (25) and (26), as illustrated for the single series expansion in equation (24). All the nonlinear terms of equations (15) have been converted to those in equation (23) for the full series expansions and the resulting relations are given in Appendix A. The definitions of the β terms (given previously in Ref. 3) have been modified to include the

effects of both full series expansions and initial imperfection presence.

The remaining nonlinear terms (those listed after equation (14)) also require modification but are of simpler form since only the first power of the rotation is involved. A redefinition of the η terms to include the effects of full series expansions and appropriate initial imperfection rotations has also been made.

In this manner, the validity of the strain-displacement equations (equations (5) of Ref. 3 and equations (15) of this report) and the equilibrium equations (equations (8) of Ref. 3) is retained. Introduction of full series expansions and initial imperfections in the nonlinear terms was accomplished by a redefinition of the β and η terms, which are the nondimensional coefficients for the nonlinear terms in the strain-displacement and equilibrium equations, respectively.

D. SUBROUTINE MODIFICATIONS

Introduction of the full series into the program was implemented by use of negative index numbers for the new sine or cosine portions of the expansions (the terms with the superscript ^{2}). With the requirement that the axisymmetric harmonic (n=0) be included, the program was expanded to handle a maximum of 25 modes, i.e., the zeroeth harmonic (n=0) plus twelve other harmonics (positive and negative). Each positive index and each negative index will be referred to as a mode, i.e., the harmonic n=5 has two modes, -5

and +5. A typical correspondence between harmonic, mode and Fourier index is given in Table I. Harmonic zero (n=0) and any combination of up to twelve harmonics may be included in the analysis. The positive values of NN in Table I are the modes analyzed in SATANS and SATANS-I. The negative values of NN are those modes added to incorporate the full series expansions in SATANS-II. Note that n=0 is the first mode, and that the negative value is always the first mode of a pair of modes. This is a requirement.

TABLE I

HARMONIC, MODE AND FOURIER INDEX CORRESPONDENCE

HARMONIC n	MODE	FOURIER INDEX	FORTRAN ASSIGNMENT STATEMENT IN SUBROUTINE PLOAD OR TLOAD
0	1	0	NN(1)=0
1	2	-1	NN(2)=-1
	3	+1	NN(3)=+1
2	4	-2	NN(4)=-2
	5	+2	NN(5)=+2
5	6	-5	NN(6)=-5
	7	+5	NN(7)=+5
.	.	.	.
.	.	.	.
.	.	.	.
18	24	-18	NN(24)=-18
	25	+18	NN(25)=+18

Seven subroutines of SATANS-I required changes to incorporate the full trigonometric series expansions and the initial imperfections.

Subroutine PHIBET calculates the ϕ's and carries out the summation procedures for calculating the β nonlinear terms. Since extensive modification was required, the subroutine was altered such that those sections handling the full trigonometric expansions and imperfections are entered only if required, i.e., if a full series analysis is necessary.

Subroutine TEAETA calculates the inplane forces and the η nonlinear terms. It was also extensively modified, again, utilizing new sections only if required.

The subroutine MODES, a highly complex procedure for determining modal coupling required minimal alteration since each unique mode number is stored at every, or every other storage location, depending on use of SATANS-I or SATANS-II, respectively. Each unique harmonic is accessed, then, by an appropriate step size change in the controlling loops.

The subroutine OUTPUT required the addition of a section to include the new half of the trigonometric series in the generation of a summed solution at a specified meridian.

The subroutines POLE, INLPOL and FNLPOL compute the solution at an initial or a final pole. Due to the computation of pole conditions based on n, which now has positive and negative values, these routines were modified

accordingly. Effectively, the only change in the equations presented in Appendix D of Ref. 3 involve a replacement of n by -n where appropriate.

A complete listing of these seven modified subroutines is given in Appendix F.

E. SATANS-II

To modify SATANS-I to handle completely arbitrary loading and initial imperfections, the following steps must be performed.

1. Replace subroutines PHIBET, TEAETA, MODES, OUTPUT, POLE, INLPOL and FNLPOL with those in Appendix F.

2. Replace the affected COMMON statements in Appendices D and E with those listed in Appendix F.

3. Add subroutine IMPERF, also listed in Appendix F. This subroutine contains the values of the imperfection rotation coefficients, $\overline{\phi}_s$ and $\overline{\phi}_\theta$. Instructions for actual preparation of these coefficients will be discussed in detail in section V. If imperfections are not to be considered, the $\overline{\phi}_s$ and $\overline{\phi}_\theta$ are never accessed.

4. In Appendix E, make the following additions after the card indicated:

Subroutine	After Card Number	Insert
DYNAMC	00002270	CALL IMPERF
STATIC	00002230	CALL IMPERF
PMATRX	00001100	IF(N(MN).LT.0)P(2,1,IJ)=-1.0
PMATRX	00002000	IF(N(MN).LT.0)ZFP1(2,2)=1.0
SATANS	00000380	READ(5,108)JUMP,MPERFS

SATANS	00001400	108 FORMAT(2I5)

5. In Appendix E, modify the following cards:

Subroutine	Card Number	Should Read
DYNAMC	00000800	DIMENSION VBAR(25),AVB(25)
PMATRX	00000900	NN=IABS(N(MN))
PMATRX	00001640	NN=IABS(N(MN))

6. In Appendix C, modify the data preparation instructions by adding a data card after card #4 according to the following format:

Card	Column	Format	Item	Interpretation
4.1	1-5	I5	JUMP	1 = single series expansion
				2 = double series expansion
4.1	6-10	I5	MPERFS	0 = no imperfections
				1 = imperfections

An initial imperfection solution requires the use of full series in the analysis. Specification of the parameter JUMP indicates whether the solution will be as in SATANS-I or will utilize the full trigonometric series expansions of SATANS-II. Specification of the parameter MPERFS indicates whether the analysis will include initial imperfections. Note that if MPERFS equals 1, JUMP must equal 2, but if JUMP equals 2, MPERFS may be either 1 or 0.

7. Add imperfection data just prior to the last card of the data deck. This data is to be placed in arrays PHIXB and PHITB.

F. USE OF SATANS-II

After completing the changes in section IV.E, SATANS-II exists as a program capable of performing all the functions of SATANS-I with the following restrictions:

1. The number of meridional stations cannot exceed 100, (KMAX < 101).

2. The number of modes cannot exceed 25, (MNMAX < 26).

3. The product of meridional stations and modes cannot exceed 951, (KMAX*MNMAX < 952).

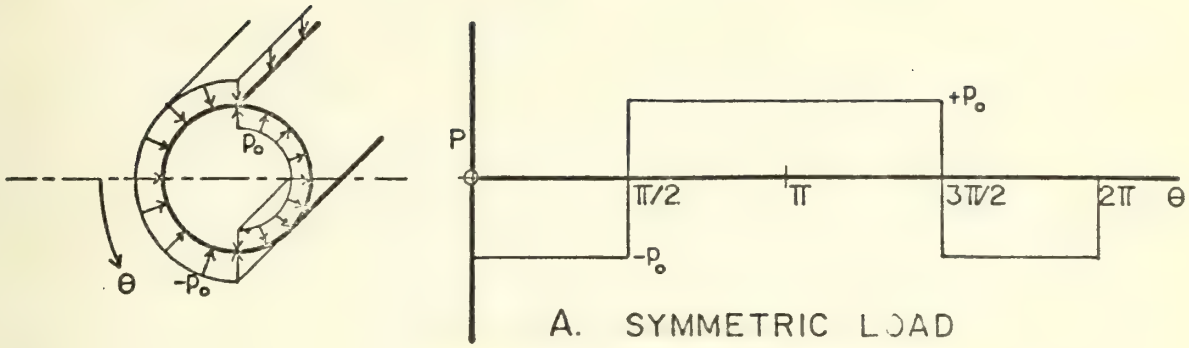
4. Plotting ability is not available for the imperfection input data.

Specification of harmonics to be considered (accomplished in subroutine PLOAD or TLOAD) must include the axisymmetric harmonic (n=0) and both negative and positive entries for any other harmonic, as indicated in Table I.

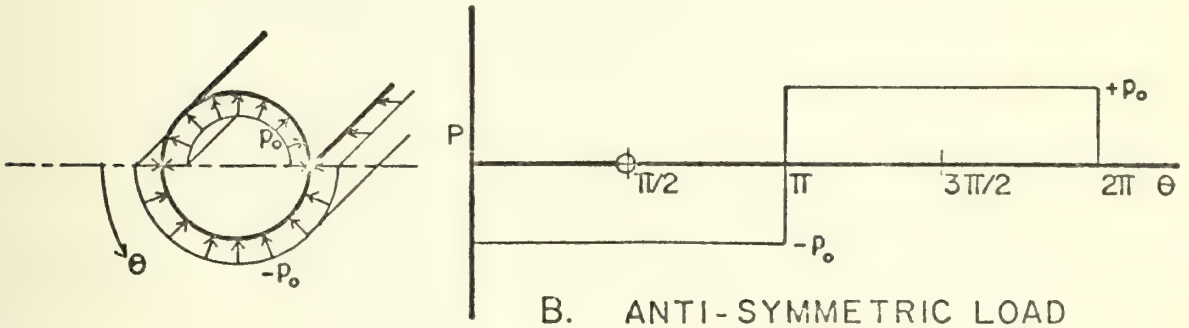
The instructions for the preparation of the input data are given in Appendix C. SATANS-II requires 426 k-bytes of main core on an IBM-360/67 computer when run under OS/MVT and when compiled by the FORTRAN-H compiler at optimization level 2. Execution times vary widely from four minutes for a 35 station and 3 mode analysis, up to 30 - 60 minutes for a 35 station and 25 mode analysis.

G. VALIDITY OF SATANS-II

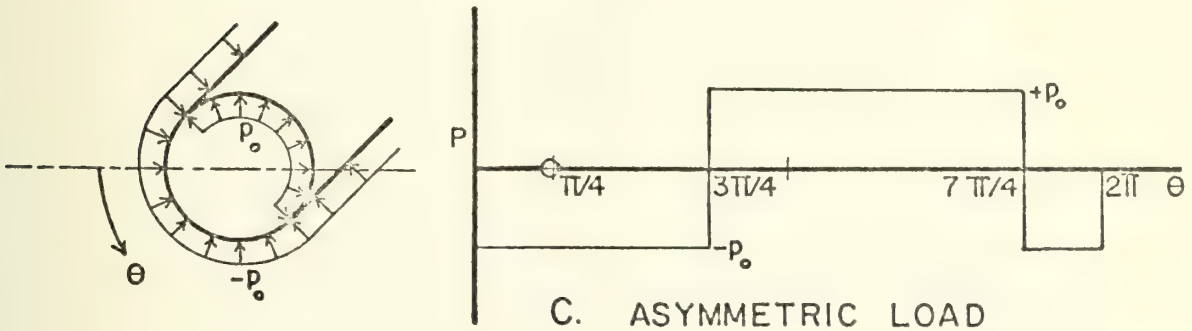
SATANS-II was checked for its ability to obtain a correct solution using the full series expansions. This check was made using a clamped cylindrical shell subjected to pressure loadings as described in Figure 5. A solution was



A. SYMMETRIC LOAD



B. ANTI-SYMMETRIC LOAD



C. ASYMMETRIC LOAD

⊕ = SOLUTION COMPARISON POINTS

SATANS-III CHECK LOADINGS

FIGURE 5

obtained for the loading shown in Figure 5A with SATANS-I and the summed solution was printed at meridian $\theta = 0^{o}$. The cylinder was then analyzed by SATANS-II for the following three orientations:

1. Loading defined as in Figure 5A - a symmetric load.

2. Loading defined as in Figure 5B - an anti-symmetric load. In this case, the summed solution was printed at meridian $\theta = 90^{o}$.

3. Loading defined as in Figure 5C - a general load. In this case, the summed solution was printed at meridian $\theta = 45^{o}$.

Load definitions for the four check runs appear in Table II. In each case, the solution obtained by SATANS-II

TABLE II

FOURIER COEFFICIENTS FOR CHECK LOADINGS

MODE	FOURIER COEFFICIENTS OF APPLIED LOADING		
	SYMMETRIC	ANTI-SYMMETRIC	NONSYMMETRIC
0	0.0	0.0	0.0
-1	0.0	-63.66	-45.015
+1	-63.66	0.0	-45.015
-3	0.0	+21.22	+15.005
+3	+21.22	0.0	-15.005
-5	0.0	-12.73	+9.003
+5	-12.73	0.0	+9.003
-7	0.0	+9.095	-6.431
+7	+9.095	0.0	+6.431

Note: the coefficients are based on p_o = 50 psi.

agreed with the SATANS-I solution[2] through four significant digits.

[2]This reference to solution includes the load-deflection history, the ultimate nonconvergence level, and the displacement, force and moment distributions.

V. IMPERFECT SHELL ANALYSIS

A. PREPARATION OF THE IMPERFECTION DATA

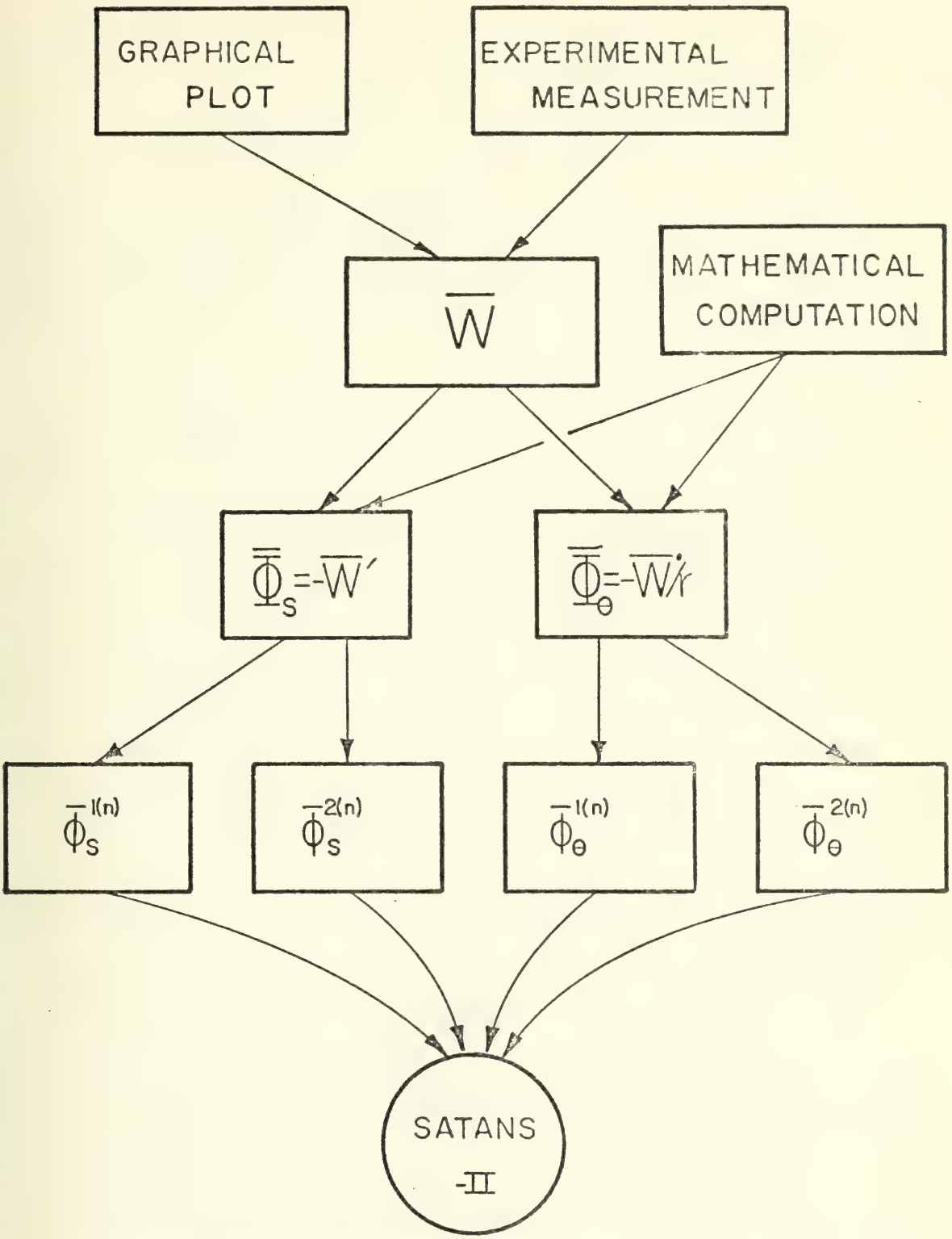
With the developments of section IV implemented, SATANS-II requires only the Fourier coefficients of the imperfection data to conduct a nonlinear analysis. Recall from equation (26) that these Fourier coefficients, $\bar{\phi}_s$ and $\bar{\phi}_\theta$ are determined by knowledge of the actual imperfection rotations, $\bar{\Phi}_s$ and $\bar{\Phi}_\theta$. Recall also from equation (22) that the imperfection rotations are given as

$$\bar{\Phi}_s = -\bar{W}'$$
$$\bar{\Phi}_\theta = -\bar{W}^\bullet/r \tag{27}$$

This means that the imperfection rotation Fourier coefficients can be determined by knowledge of the initial normal deflection deviation, $\bar{W}$. This imperfection information can be obtained from a mathematical description, a graphical plot or from actual experimental measurement. A simplified data generation flow chart is presented in Figure 6.

1. Data Generation by Mathematical Description

In a note on imperfect shallow spherical cap buckling, Kao [Ref. 13] analyzes a cap subjected to a mathematically defined dimple imperfection existing over one-quarter of the shell. The dimple is given by



SIMPLIFIED IMPERFECTION DATA GENERATION SCHEME

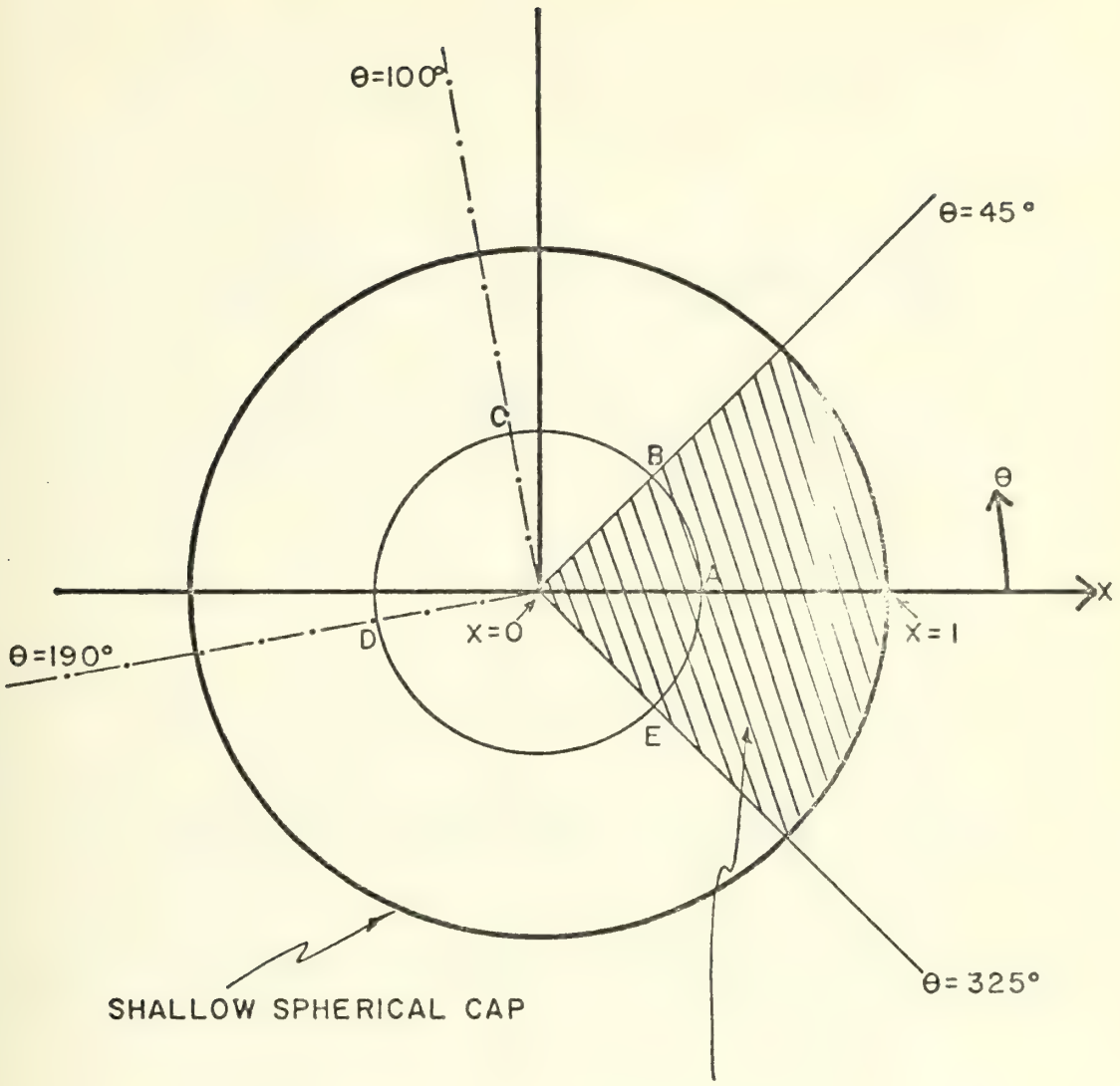
FIGURE 6

$$\bar{W} = 64(s/L)^3(1-s/L)^3 h\delta \cos 2\theta, \quad -45^\circ < \theta < 45^\circ$$

$$\bar{W} = 0, \quad 45^\circ < \theta < 325^\circ \tag{28}$$

where L is the total length of the shell meridian and where s/L ranges from 0 at the pole to 1 at the edge, and δ is an intensity factor, $0 \leq \delta \leq 1$. The geometry of this problem is shown in Figure 7. Kao used a two-dimensional finite difference mesh of eight meridional and 16 circumferential stations in his analysis. A numerical model of, for example, 35 meridional stations and 25 modes would yield a SATANS-II solution of considerable accuracy. The discrete initial normal deflection deviations $\bar{W}$ at each mesh point need not be computed, as partial differentiations of equation (28) give expressions directly solvable for discrete values of $\bar{\Phi}_s$ and $\bar{\Phi}_\theta$ for each s and θ. Imperfection rotations at locations exterior to the affected area ($-45^\circ < \theta < 45^\circ$) are zero. To illustrate, the meridional imperfection rotations that exist along the line ABCDEA of Figure 7 would appear as curve A of Figure 8. The discrete values generating this curve can easily be expanded into a full Fourier series and the resulting coefficients $\bar{\phi}_s^{(n)}$ inserted into SATANS-II for the analysis.

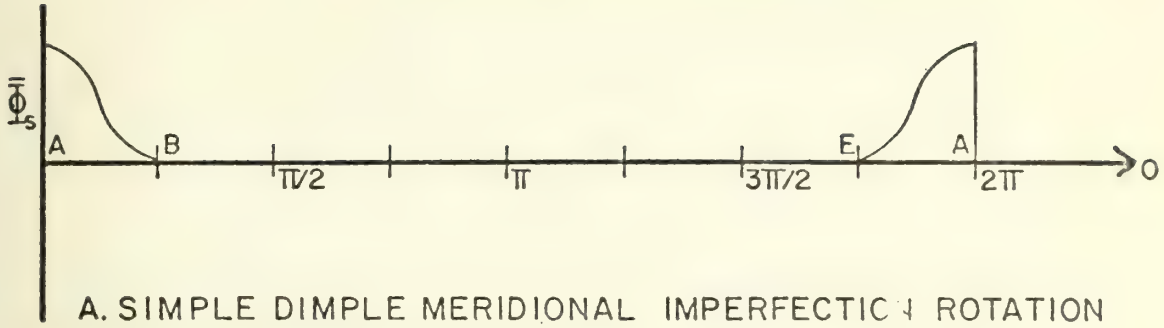
The imperfection of Figure 7 is symmetric about the meridian $\theta = 0^\circ$. As an example of a more complex imperfection, another dimple of the same form could be located in the region defined by $100^\circ < \theta < 190^\circ$. In this asymmetric case, the meridional imperfection rotations along line



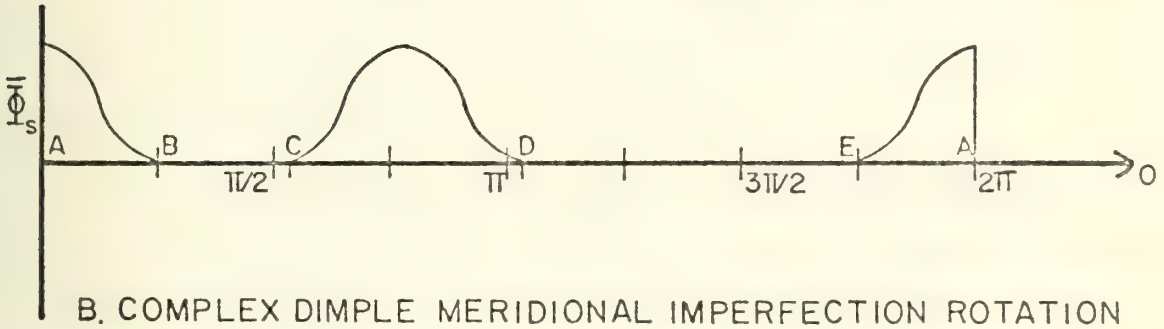
IMPERFECT QUARTER WITH
DIMPLE IMPERFECTION GIVEN BY
$\overline{W}/h = 64X^3(1-X)^3\, d \cos 2\theta$
WHERE $X = s/L$

MATHEMATICAL IMPERFECTION GENERATION

FIGURE 7



A. SIMPLE DIMPLE MERIDIONAL IMPERFECTION ROTATION



B. COMPLEX DIMPLE MERIDIONAL IMPERFECTION ROTATION

MERIDIONAL IMPERFECTION ROTATION REPRESENTATION

FIGURE 8

ABCDEA of Figure 7 would now appear as curve B of Figure 8. The discrete values generating this curve can also be easily expanded into the full trigonometric series and its coefficients inserted into SATANS-II.

2. Data Generation by Graphical Plots

Graphical representations of $\overline{W}$ can be easily constructed and the patterns are near limitless in design or location. Theoretical or actual creases, furrows or dents of any shape need only be graphed, the discrete imperfection displacements determined, the discrete imperfection rotations $\overline{\Phi}_s$ and $\overline{\Phi}_\theta$ computed via a finite differencing technique, the Fourier expansions performed and the resulting coefficients inserted into SATANS-II.

3. Data Generation by Experimental Measurement

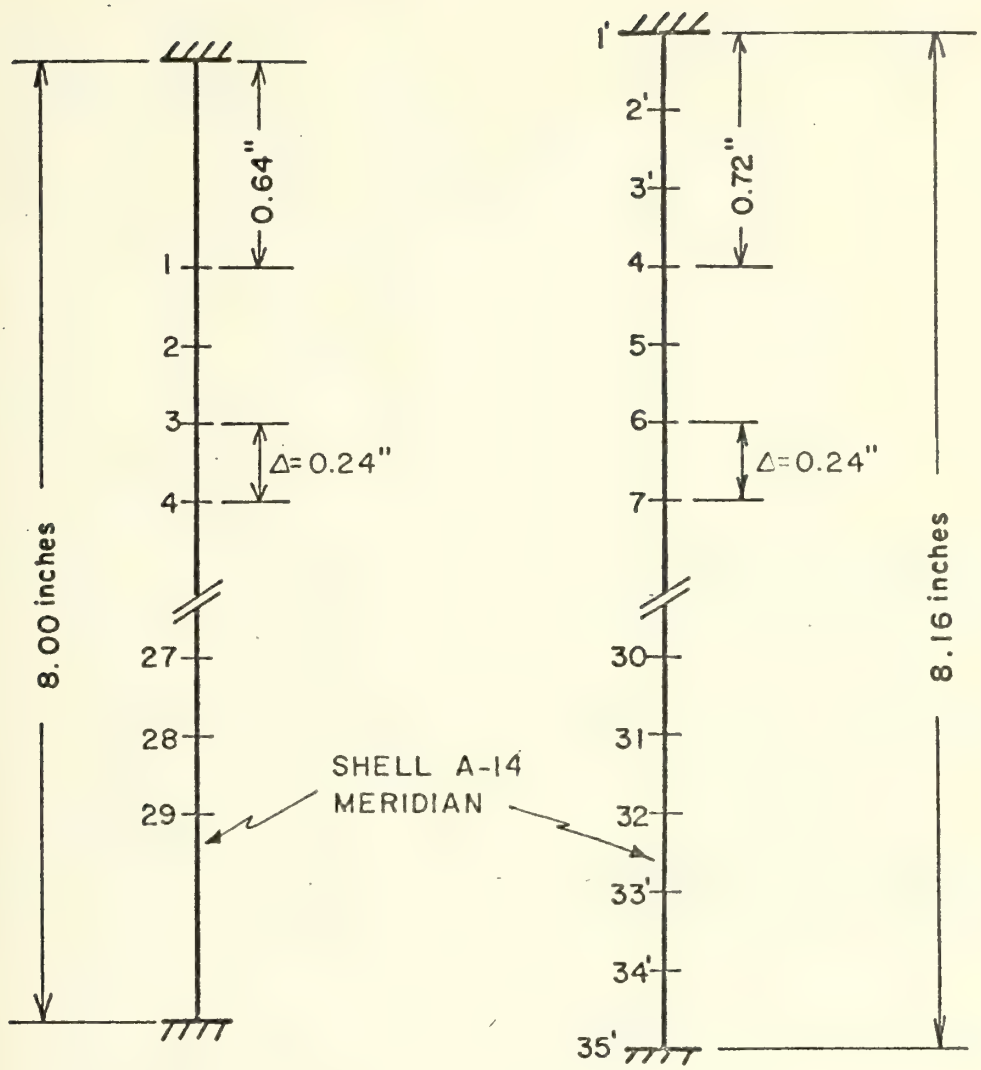
Arbocz and Babcock [Ref. 6] have developed a means of mapping the topography of circular cylindrical shells, and can thus describe the deviations of an actual shell. The procedure, described in detail in Ref. 6, produces imperfection data in the form of nondimensional normal displacements from a perfectly circular cylinder. For the sample shells considered here, 49 data points were taken around each circumference and 29 equally-spaced circumferential traces were made. The first and last trace were 0.64 inches from the ends of the shell and the spacing between traces was 0.24 inches. Pertinent geometrical and material data concerning the shells of interest here is contained in Table III.

TABLE III

IMPERFECT SHELL OVERALL CHARACTERISTICS

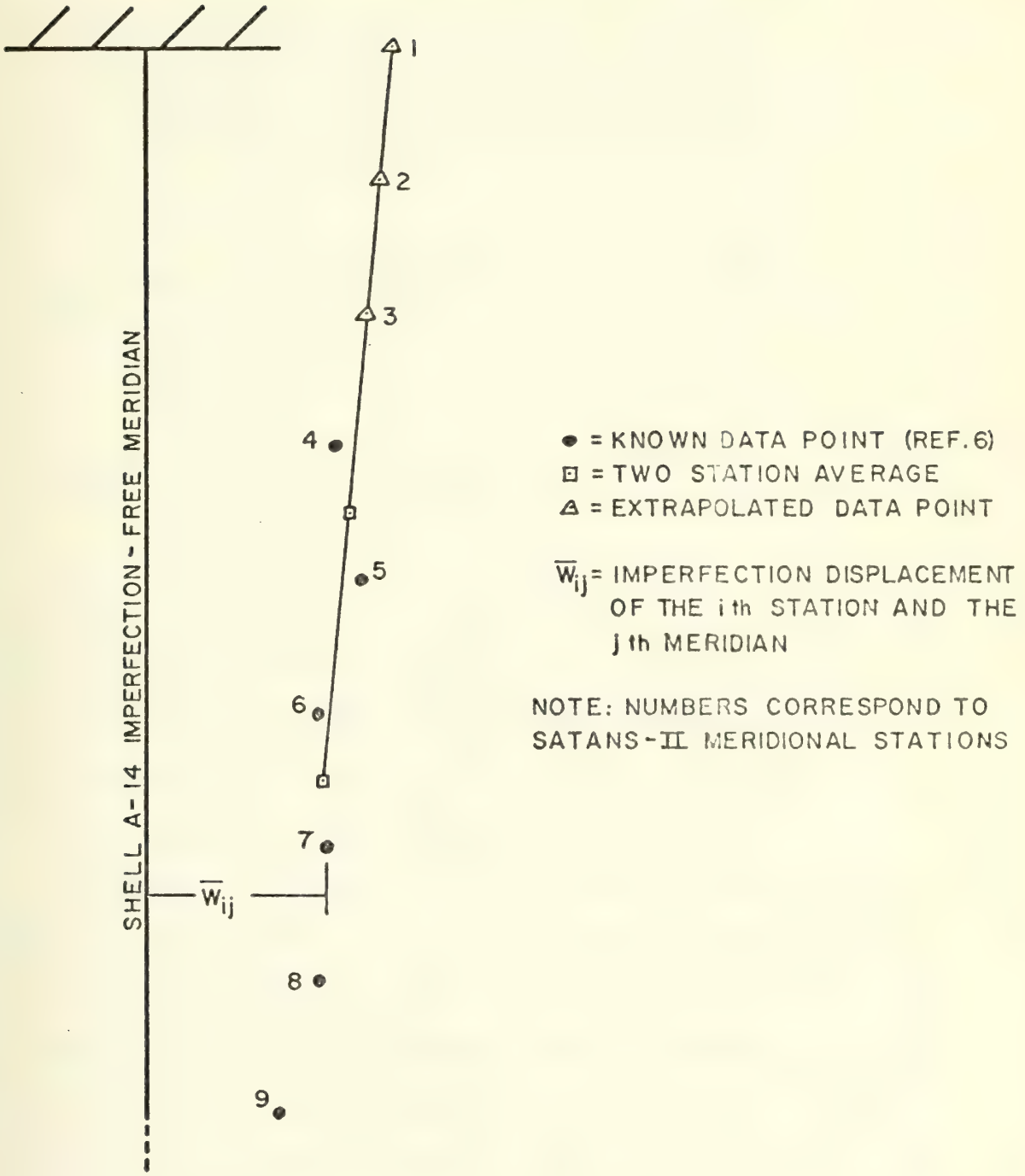
SHELL DESIGNATION	A-13	A-14
Radius	4.00"	4.00"
Length	8.00"	8.00"
Young's Modulus x 10^6, E	15.10 psi	15.80 psi
Wall thickness, h	0.00444"	0.00437"
Number of axial traces	29	29
Axial increment, Δ	0.24"	0.24"
Number of data points	1421	1421

To prepare the imperfection data on shell A-14 for use by SATANS-II, the nondimensional deviations were first dimensionalized by the wall thickness. To retain the original axial increment of 0.24 inches between circumferential traces and to make the data compatible with a 35-station meridian, the shell length was increased by 0.08 inches at each end, or a total of 2% (See Figure 9). This required three additional traces at each end of the shell corresponding to stations 1, 2 and 3, and 33, 34 and 35. Data for the additional six circumferential traces were generated via the modified linear extrapolation procedure illustrated in Figure 10. At each of the 1,715 data points, the meridional and circumferential rotations were computed



CIRCUMFERENTIAL TRACE NUMBER AND
MERIDIONAL STATION NUMBER CORRESPONDENCE

FIGURE 9



DATA POINT EXTRAPOLATION TECHNIQUE FOR $\overline{W}$

FIGURE 10

using central finite differencing in the appropriate direction. After nondimensionalization according to the SATANS-II scheme, each circumferential vector of 48 meridional and 48 circumferential imperfection rotations was expanded via a periodogram into a Fourier series according to

$$(E_o/\sigma_o)\overline{\Phi}_s = \overline{\phi}_s^{(0)} + \sum_1^{24} \overline{\phi}_s^{c(n)} \text{Cos } n\theta + \sum_1^{24} \overline{\phi}_s^{s(n)} \text{Sin } n\theta \quad (29)$$

and

$$(E_o/\sigma_o)\overline{\Phi}_\theta = \overline{\phi}_\theta^{(0)} + \sum_1^{24} \overline{\phi}_\theta^{c(n)} \text{Cos } n\theta + \sum_1^{24} \overline{\phi}_\theta^{s(n)} \text{Sin } n\theta \quad (30)$$

At this point, the Fourier coefficients $\overline{\phi}_s$ and $\overline{\phi}_\theta$ are of proper form for direct insertion into SATANS-II. The computer program which performed the actual imperfection data processing for shell A-14 is given in Appendix G.

4. Selection of Important Harmonics

Since the number of meridional stations was 35, the program maximum of 25 modes can be used in the analysis.[3] Note, however, that in equations (29) and (30), there are 49 modes describing the imperfections. Therefore, a selection of a maximum of 25 imperfection modes must be made. The harmonics to be considered in the analysis should include those critical to the shell and those significant in the imperfection. The description of the harmonic

[3]Recall the restrictions given in section IV.F.

selection process for shell A-14 is given in a later section.

B. BOUNDARY CONDITIONS

The testing configuration used by Arbocz and Babcock on the axially loaded cylinder is shown in Figure 11. The ends of the shell are attached to an end ring and a load cell with cerrelow, a low melting-point alloy. The ends of the shell are believed to be clamped so that

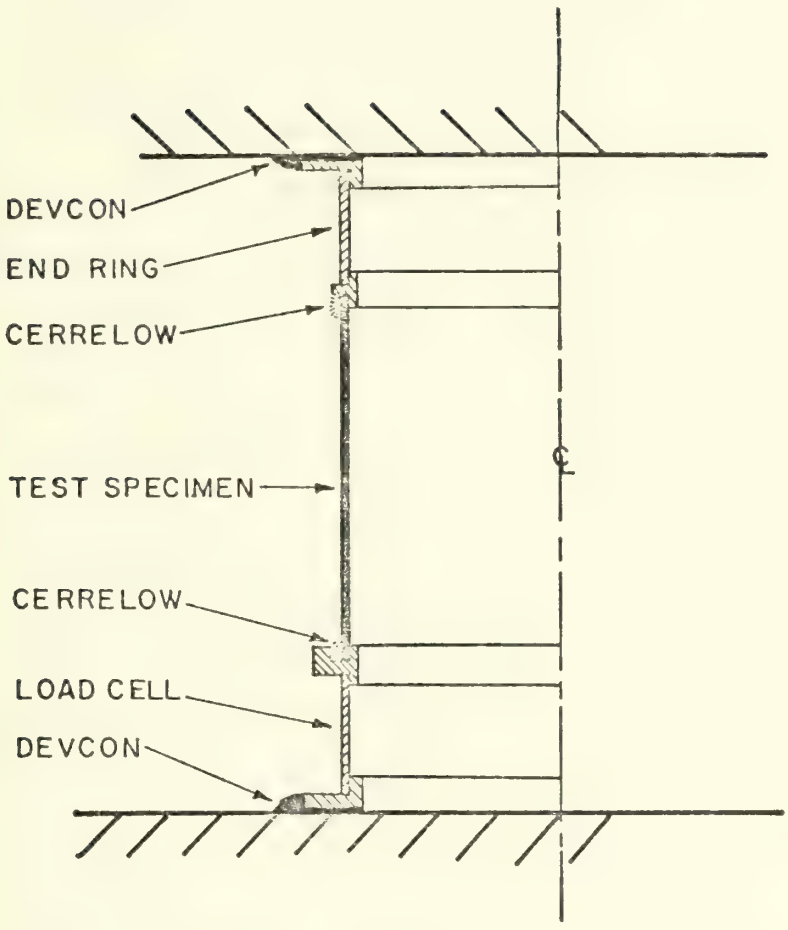
$$W = \Phi_s = V = 0 \tag{31}$$

The axial load on the test specimen was applied by means of four screws located at the corners of the square end plates. At each load increment, the screws were adjusted to give as uniform a circumferential load distribution as possible.

In all versions of SATANS, the boundary conditions are represented by the matrix equation

$$\left[\Omega\right] \begin{Bmatrix} N_s \\ N_s \\ Q_s \\ \Phi_s \end{Bmatrix} + \left[\Lambda\right] \begin{Bmatrix} U \\ V \\ W \\ M_s \end{Bmatrix} = \{\ell\} \tag{32}$$

Boundary conditions can be specified on N_s or U, $N_{s\theta}$ or V, Q_s or W, and Φ_s or M_s by prescribing input values for the matrices $\left[\Omega\right]$ and $\left[\Lambda\right]$. There is one equation (32) for each boundary of the shell. In their present form, SATANS-I



CYLINDRICAL SHELL TESTING CONFIGURATION

FIGURE 11

and SATANS-II assume the column matrix $\{\ell\}$ as zero for all harmonics except the axisymmetric harmonic.

As was mentioned earlier, boundary condition discrepancy between mathematical model and physical experiment is cited as a major contributor to thin shell buckling load disagreement. For this reason, three interpretations of Figure 11 were made. The cylinder was represented as experiencing an axisymmetrically applied force N_s on both ends (condition A), an axisymmetrically applied force N_s on one end and a meridional deflection restraint on U at the other (condition B), and an axisymmetric incremental end shortening U at one end with meridional deflection U restrained at the other (condition C).[4] These three conditions were investigated on the clamped cylinder, and on a simply supported cylinder for further comparisons and validity checks. Mathematical assignments for boundary conditions A, B and C are shown in Table IV.

C. ANALYSIS

1. Buckling of the Thin Perfectly-Circular Cylinder

A necessary part of this attempt to develop a program for predicting the buckling load of imperfect shells is the establishment of the program's ability to predict the buckling load of perfect shells. Consequently, this section compares the results of a SATANS-II analysis of the thin

---

[4] It is believed that boundary condition C best represents the actual boundary conditions in Figure 11.

TABLE IV

BOUNDARY CONDITION SPECIFICATIONS

SUPPORT	BOUNDARY CONDITION	BOUNDARY	N_s or U	$N_{s\theta}$ or V	Q_s or W	Φ_s or M_s
Clamped	A	Upper	$N_s^{(n)}=-N_o$	$V^{(n)}=0$	$W^{(n)}=0$	$\Phi_s^{(n)}=0$
		Lower	$N_s^{(n)}=-N_o$			
	B	Upper	$N_s^{(n)}=-N_o$			
		Lower	$U^{(n)}=0$			
	C	Upper	$U^{(n)}=U_o$			
		Lower	$U^{(n)}=0$			
Simple	A	Upper	$N_s^{(n)}=-N_o$	$V^{(n)}=0$	$W^{(n)}=0$	$M_s^{(n)}=0$
		Lower	$N_s^{(n)}=-N_o$			
	B	Upper	$N_s^{(n)}=-N_o$			
		Lower	$U^{(n)}=0$			
	C	Upper	$U^{(n)}=U_o$			
		Lower	$U^{(n)}=0$			

Note: When $N_s^{(n)}$ or $U^{(n)}$ are specified nonzero, they are set to zero for $n > 0$.

cylinder without imperfections with the bifurcation buckling analysis presented by Almroth [Ref. 14]. Almroth's study deals with the influence of edge constraints on the critical load of cylindrical shells of the type used by Arbocz and Babcock. Equation (33) is given by Almroth for the axisymmetric displacement shape of the simply supported cylinder (boundary condition A) prior to buckling.

$$W = r\left(2\nu\gamma\bar{N}_s + \bar{p}\right)\left(1 + A_1 \mathrm{Sin}(a_1 s)\mathrm{Sinh}(a_1 s) + A_2 \mathrm{Cos}(a_2 s)\mathrm{Cosh}(a_2 s)\right) \tag{33}$$

where

$$a_1 = \kappa\left(1 + \bar{N}_s\right)^{\frac{1}{2}}$$

$$a_2 = \kappa\left(1 - \bar{N}_s\right)^{\frac{1}{2}}$$

$$\kappa = \left(2\gamma\right)^{-\frac{1}{2}}$$

$$\gamma = h/r/\left(12(1 - \nu^2)\right)^{\frac{1}{2}}$$

$$\bar{N}_s = -N_s/\left(2\gamma Eh\right)$$

$$\bar{p} = (pr)/(Eh)$$

$$A_1 = -\left(\frac{\left(\{1 - \bar{N}_s^2\}^{\frac{1}{2}} \mathrm{Sin}(\alpha_1)\mathrm{Sinh}(\alpha_2) + \bar{N}_s \mathrm{Cos}(\alpha_1)\mathrm{Cosh}(\alpha_2)\right)}{\left(1 - \bar{N}_s^2\right)^{\frac{1}{2}}\left(\mathrm{Cosh}^2(\alpha_2) + \mathrm{Sin}^2(\alpha_2)\right)} \right)$$

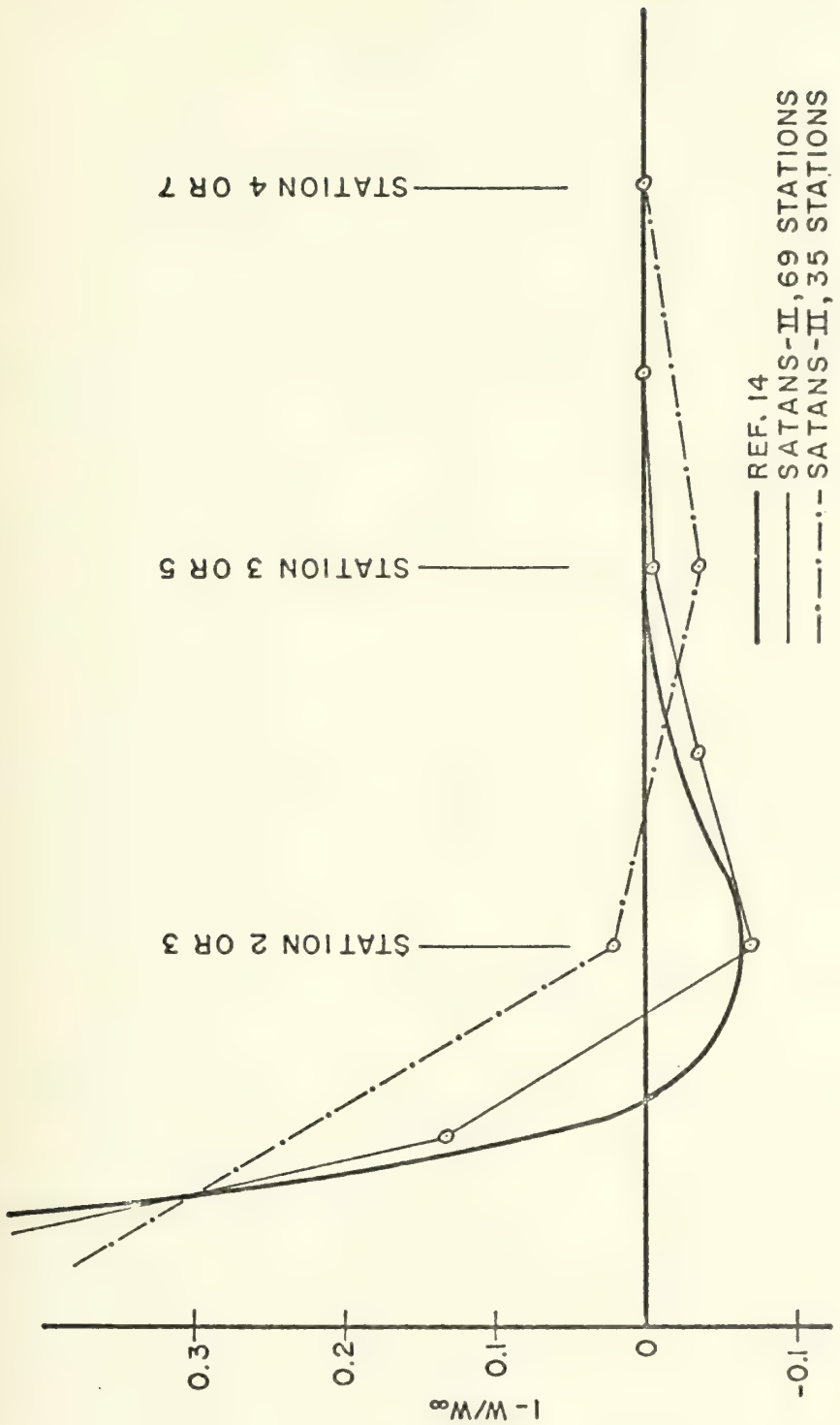
$$A_2 = -\left(\frac{\left(\{1 - \bar{N}_s^2\}^{\frac{1}{2}} \mathrm{Cos}(\alpha_1)\mathrm{Cosh}(\alpha_2) - \bar{N}_s \mathrm{Sin}(\alpha_1)\mathrm{Sinh}(\alpha_2)\right)}{\left(1 - \bar{N}_s^2\right)^{\frac{1}{2}}\left(\mathrm{Cosh}^2(\alpha_2) - \mathrm{Sin}^2(\alpha_2)\right)} \right)$$

$$\alpha_1 = (a_1 L)/(2r)$$

$$\alpha_2 = (a_2 L)/(2r)$$

and where N_s is the axial load, p is a uniform normal pressure, h is the thickness, E is Young's Modulus, ν is Poisson's ratio, L is the length and r is the radius. This solution is referred to as the consistent prebuckling solution, as opposed to the membrane or classical buckling solution where W is a constant along the meridian. Equation (33) has been solved for shell A-14 and W is plotted as a function of s in Figure 12 for an $\bar{N}_s = 0.21904$. Also shown in Figure 12 is a plot of W versus s obtained from SATANS-II for the same $\bar{N}_s$ for both 35 and 69 meridional station analyses. Note in Figure 12 that even though only a few stations lie in the boundary zone, the trend is towards the correct solution. The displacement of the 35 station solution, however, is only a rough approximation of the actual state.

Almroth has determined the minimum bifurcation buckling load, $N_{s_{cr}}$, and the corresponding value for n for a



MEMBRANE PREBUCKLE DEFLECTION SHAPE, SHELL A-14

FIGURE 12

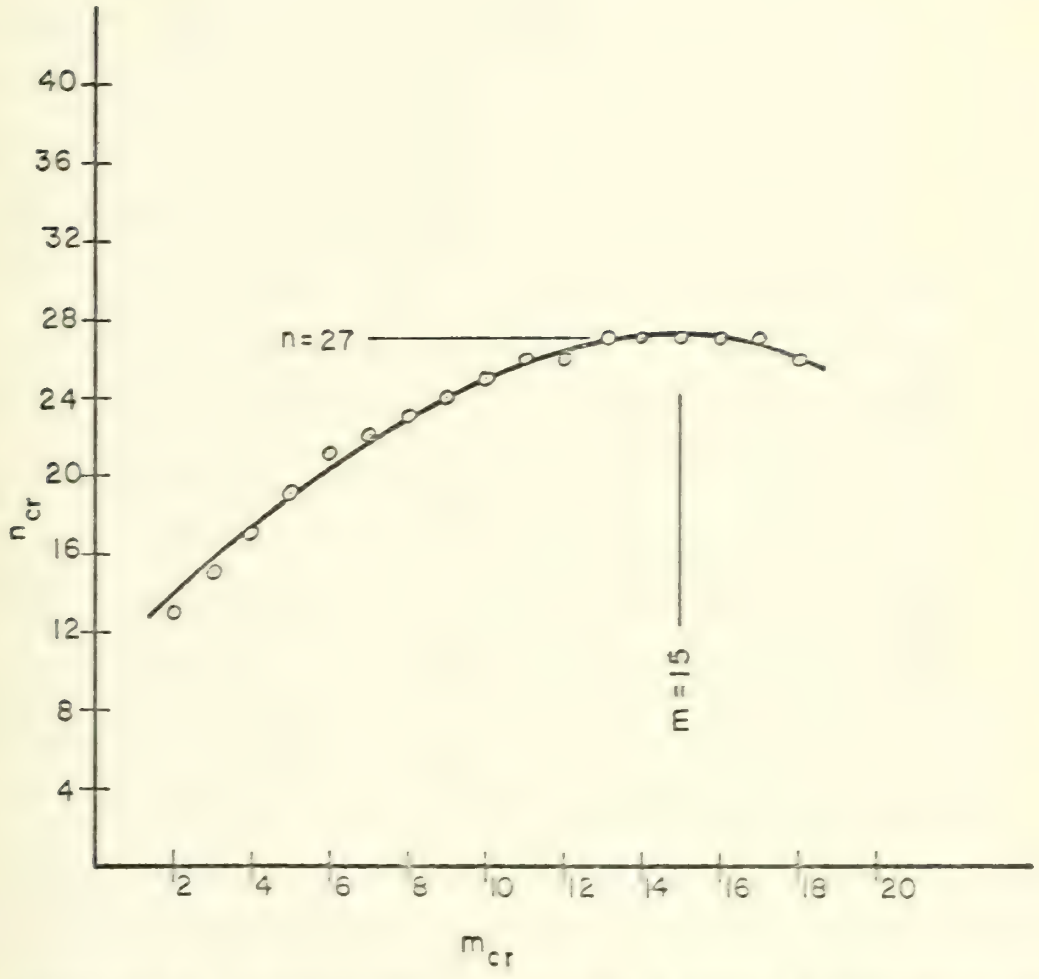
range of parameters L/r and r/h, using both the consistent prebuckling displacement W given by equation (33) and the membrane or classical prebuckling solution, where W is constant, in the Donnell equations. For the simply supported cylinder with an r/h of the same order of magnitude as shell A-14 and an L/r ≃ 0.5, the critical load is $\bar{N}_{s_{cr}}$=0.863 ($N_{s_{cr}}$=39.5 lb/in) with n=27 for the consistent analysis, and $\bar{N}_{s_{cr}}$=1.0 ($N_{s_{cr}}$=46 lb/in) with n=27 for the membrane prebuckling analysis. Thus, the consideration of the axisymmetric edge effect, which is analogous to an imperfection, reduces the buckling load by 14.1%. Parametric studies conducted by Almroth show usage of from 100 to 800 meridional stations over the half-length of the shell.

As shown in Ref. 5, use of the membrane prebuckling displacement in Sanders' equations for the simply supported cylinder simplifies the problem to an eigenvalue problem. The buckling mode is

$$W = \delta \, \mathrm{Sin}(m\pi s/L)\mathrm{Cos}(n\theta) \qquad (34)$$

The equations for the matrix elements in the eigenvalue problem are given in Appendix B. A plot of the critical values of m and n is given in Figure 13. The minimum buckling load occurs when n=27 and m=15, and corresponds to an $N_{s_{cr}}$ of 47.5 lb/in.

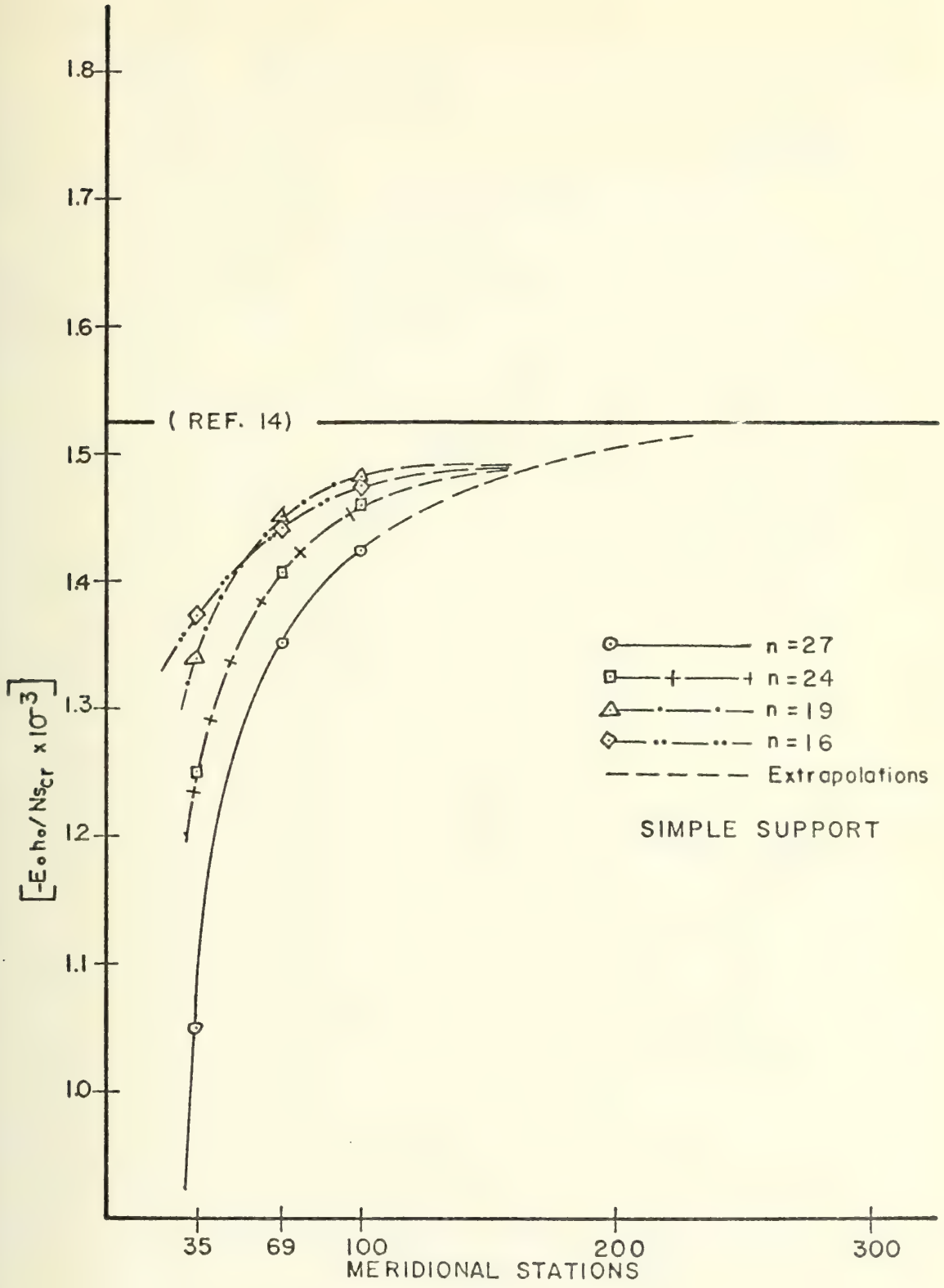
SATANS' ability to predict the minimum bifurcation load of a shell for a specified value of n has been demonstrated by Stilwell [Ref. 15]. SATANS-II analyses of shell



CRITICAL BUCKLING MODES FROM EIGENVALUE ANALYSIS, SHELL A-14

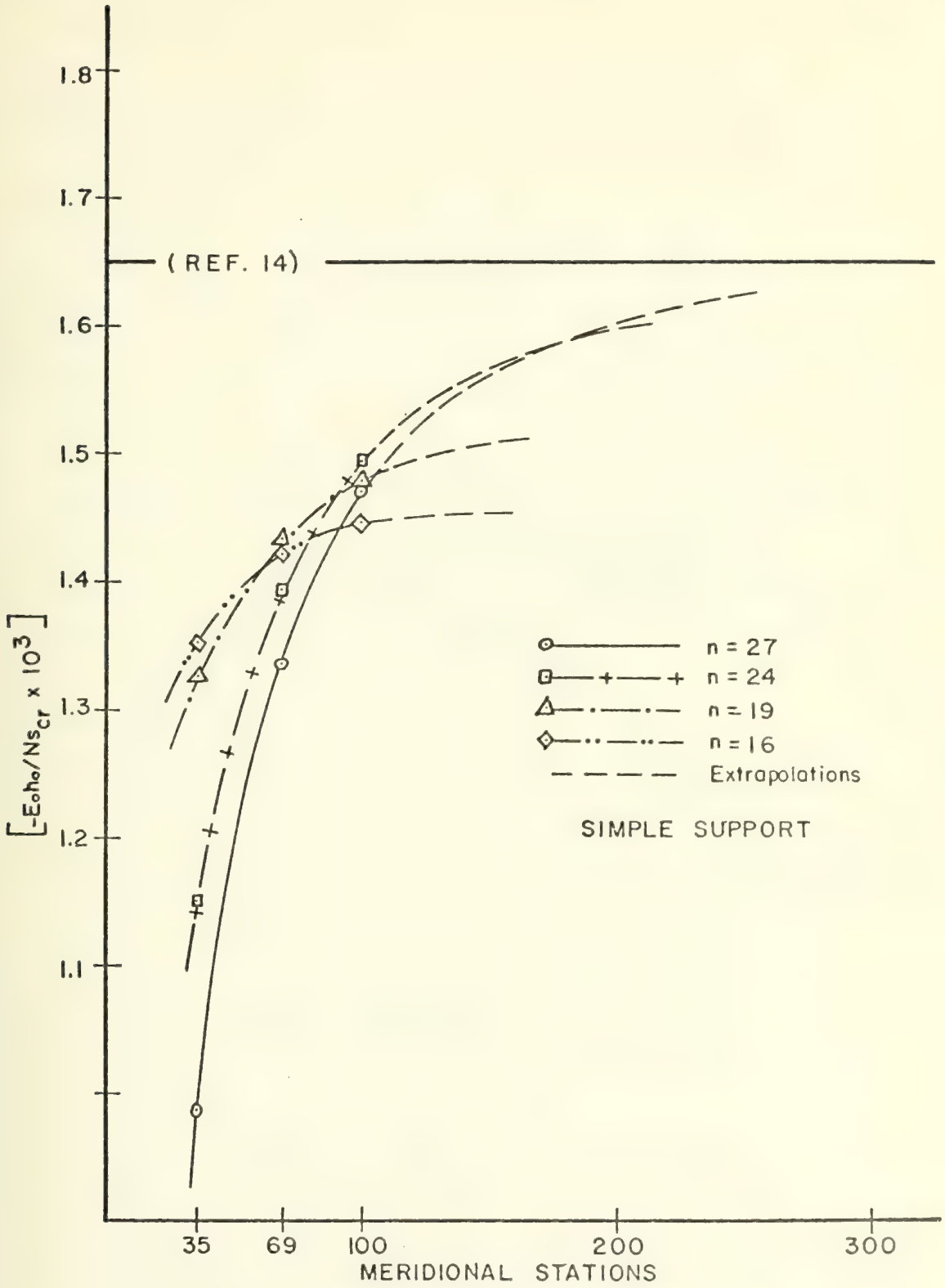
FIGURE 13

A-14 (simple support, boundary condition A) were conducted over a wide range of n for both the consistent and membrane prebuckle cases. Figures 14 and 15 give plots of the critical buckling load versus the number of meridional stations for various values of n. The dashed parts of the curves are estimated extrapolations. It is clear that as the number of stations is increased, SATANS' critical buckling load decreases and the associated harmonic increases. While the n=27 curve predicts the highest buckling load over the range of stations used in both cases, extrapolation of these curves indicates that the n=27 curve would predict the lowest buckling load if the number of stations exceeded 200 to 300. Note that the membrane prebuckle critical buckling curves of Figure 15 seem to flatten much more quickly than for the consistent case. This appears to confirm the occurence of a reduction in critical buckling load due to the presence of an axisymmetric edge restraint, as noted by Almroth. Figures 16 and 17, which present plots of the critical buckling load versus critical buckling mode of shell A-14 for the 35, 69 and 100 station analyses, also predict this lowering of buckling load due to edge restraint. the 'fanning' of curves in Figure 16 indicate an increase in the number of meridional stations will produce a lesser buckling load decrease than in Figure 17. These figures also predict a lower critical buckling load and a higher critical buckling mode with increasing meridional stations.



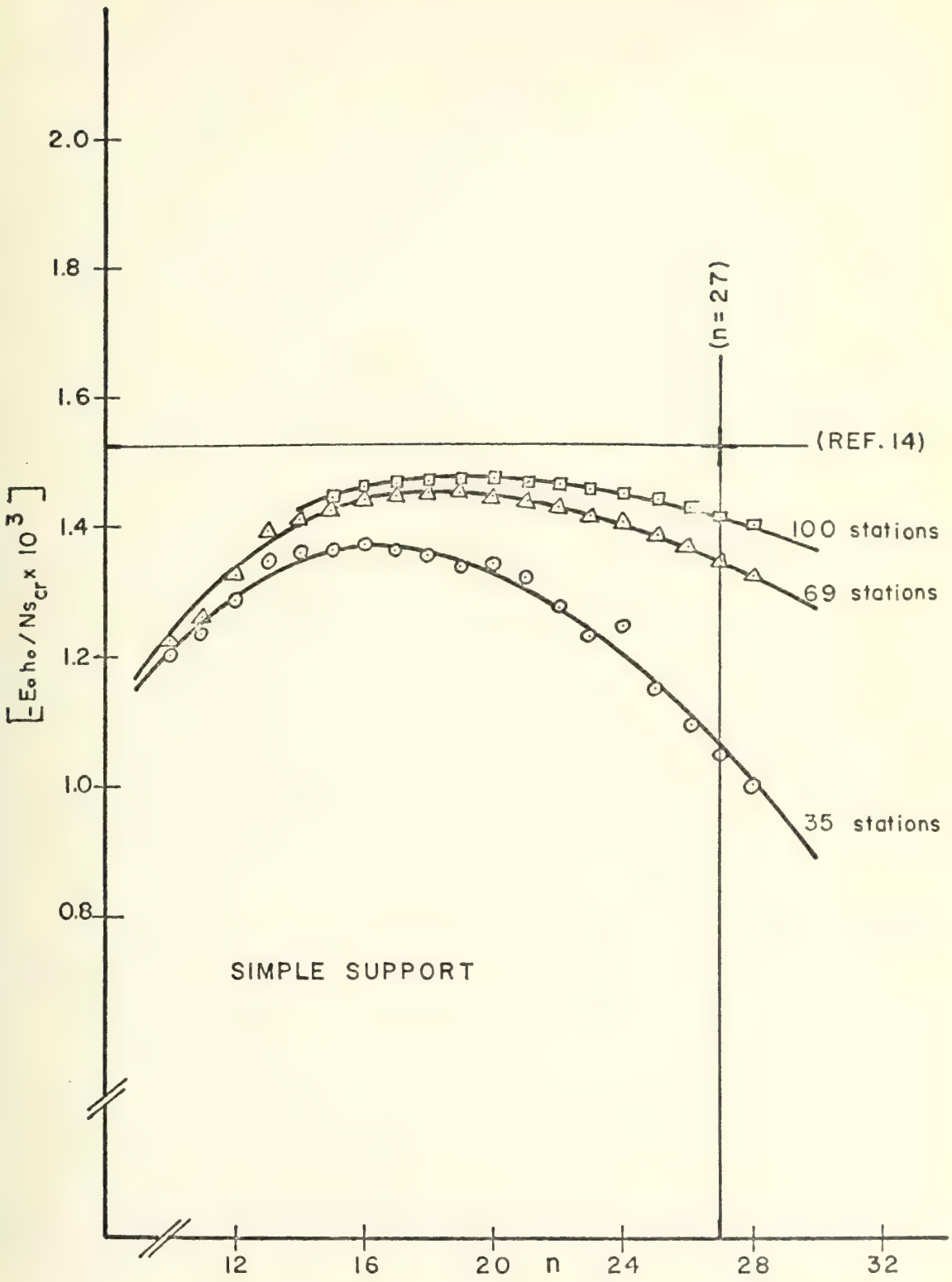
BUCKLING LOAD vs STATIONS
MEMBRANE PREBUCKLE ANALYSIS, SHELL A-14

FIGURE 14



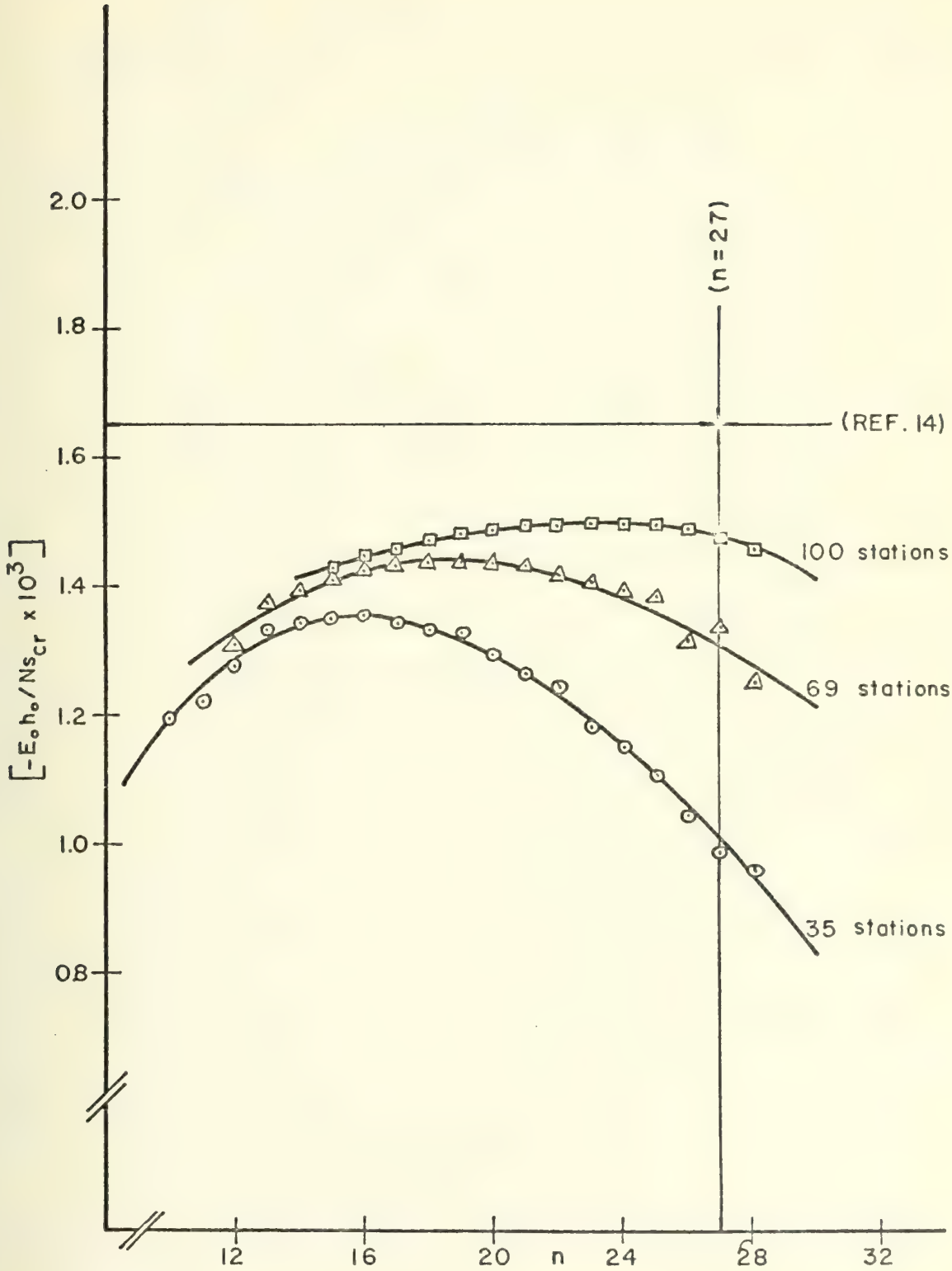
BUCKLING LOAD vs STATIONS
CONSISTENT ANALYSIS, SHELL A-14

FIGURE 15



BUCKLING LOAD vs BUCKLING MODE
MEMBRANE PREBUCKLE ANALYSIS, SHELL A-14

FIGURE 16



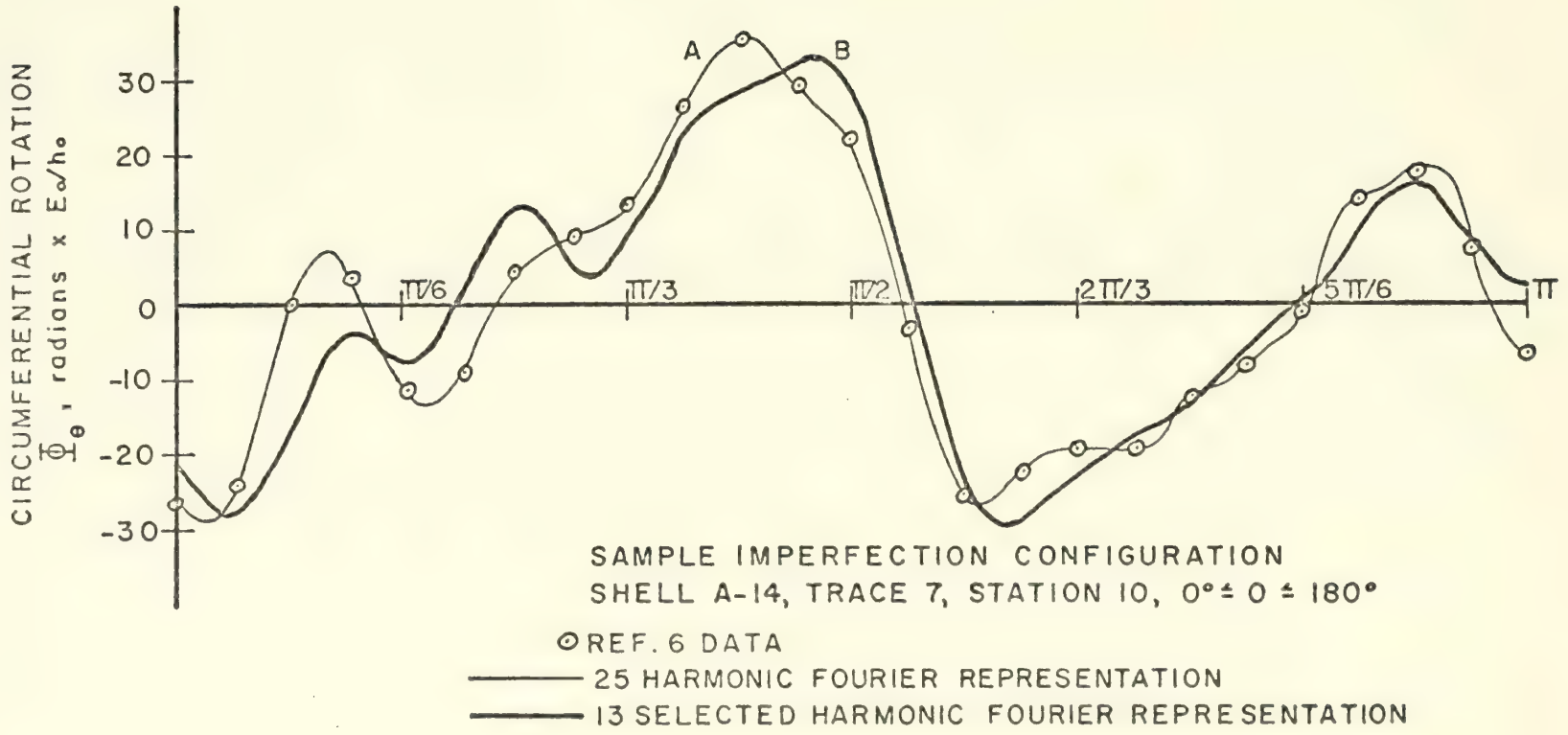
BUCKLING LOAD vs BUCKLING MODE
CONSISTENT ANALYSIS, SHELL A-14

FIGURE 17

Figures 14 through 17 indicate that, given enough stations, SATANS-II can predict a critical buckling load and mode for the thin-walled shell that agrees with Almroth's prediction. However, for the 35-station analysis, which is the analysis to be used for the imperfect shells, the critical load of 51.1 lb/in and harmonic of 16 are significantly different from the correct values. As a result of the coarse mesh used by SATANS-II in the following analysis, the critical buckling loads for the imperfect shells will be higher than the experimental values. Nevertheless, it will be shown that the introduction of a totally arbitrary, naturally-occurring imperfection pattern on the imperfection-free mathematical model of shell A-14 results in a significant reduction in critical buckling load. This is not only sufficient to demonstrate the capability of the program to perform the analyses, but it also provides an estimate of the degree of imperfection sensitivity of the shell. It will be shown that application of this estimate of sensitivity to Almroth's buckling load provides an accurate prediction of the buckling load of the imperfect shell.

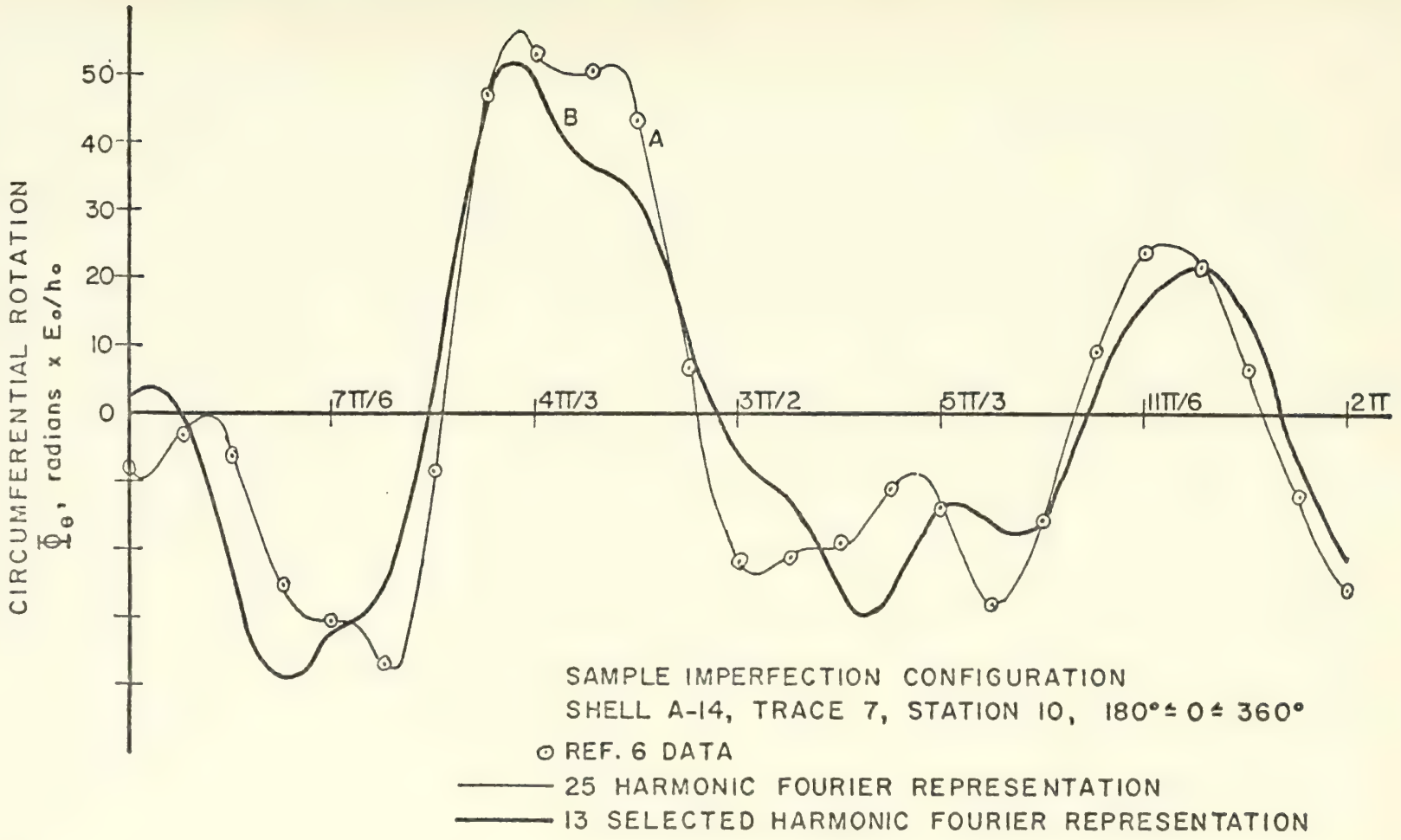
2. Introduction of Imperfections

When prepared as indicated in section V.A.2, the imperfection rotation Fourier coefficients are in a form for direct introduction into SATANS-II. Figure 18 shows the applicability of the periodogram method of expansion of the imperfection data according to equations (29) and (30).



CIRCUMFERENTIAL IMPERFECTION ROTATION REPRESENTATIONS - Φ_θ vs θ

FIGURE 18 - PLATE 1 OF 2



CIRCUMFERENTIAL IMPERFECTION ROTATION REPRESENTATION - Φ_θ vs θ

FIGURE 18 PLATE 2 OF 2

Curve A is the graph of a solution of equation (30) presenting the circumferential imperfection rotation at station 10 plotted against θ. It is seen to pass smoothly through all the data points.

Only the zeroeth harmonic (n=0) and twelve others can be accommodated by SATANS-II. In the case where the shell response and the imperfection are limited to a few major harmonics, all important harmonics can be included in the analysis. For example, the λ=6 shallow spherical cap has only the n=2 harmonic as the major harmonic of response, and the imperfection curves of Figure 8 require only a few Fourier harmonics for adequately accurate representation. Also, the few required harmonics for imperfection representation are in the vicinity of the n=2 harmonic. This problem is will within the limits of SATANS-II.

The thin-walled cylinder, however, responds in a large number of harmonics. This can be seen in Figure 16 -- many harmonics produce critical buckling loads close to the value of the minimum critical buckling load. Additionally, the imperfection requires many harmonics. Thus, as mentioned in section V.A.4, a difficult selection of harmonics for use in the analysis must be made. Six harmonics were selected on the basis of the shell response and six harmonics were selected on the basis of the imperfection characteristics. Of course, all 13 selected harmonics will contain both the imperfection characteristics and the shell response characteristics.

For the shell response harmonic selection, the critical buckling loads of shell A-14 for specified values of n, discussed in the previous section, are presented in Table V. Values were assigned to each harmonic according to the legend of Table V and six were selected based on these assignments.

For the imperfection harmonic selection, the sums of the absolute values of the Fourier coefficients for both sine and cosine terms of both meridional and circumferential imperfection rotations were tabulated and are presented in Table VI. Values were again assigned to each harmonic according to the legend of Table V, and six additional harmonics were selected.

The complete harmonic selection process for shell A-14 was accomplished by a comparison of the values assigned to each harmonic for both the imperfection and the shell. Table VII presents this comparison along with the actual harmonics selected for the analysis. Note that no overlap of important harmonics occurred, as the imperfection required harmonics seven and below and the shell required harmonics ten and above. The harmonics used with shell A-13 were n=0 and n=2,3,4,6,8,9 for the imperfection and n=10,11, 12,13,14,15 for the shell response with a simple support, and n=13,14,15,16,17,18 for the shell response with a fixed support.

Once the specific harmonics for inclusion in the analysis are determined, the appropriate $\overline{\Phi}_s$ and $\overline{\Phi}_\theta$

TABLE V

CRITICAL BUCKLING LOADS OF INDIVIDUAL HARMONICS

HARMONIC	FIXED SUPPORT			SIMPLE SUPPORT		
	BUCKLING LOAD x P_o	P/P_{cr}	ASSIGNED VALUE	BUCKLING LOAD x P_o	P/P_{cr}	ASSIGNED VALUE
2	6+			6+		
3	6+			6+		
4	5.320			6+		
5	5.120			5.720		
6	4.250			4.080		
7	2.415	2.380		2.445	2.390	
8	1.611	1.550		1.661	1.630	
9	1.221	1.180	-	1.291	1.260	
10	1.093	1.050	*	1.157	1.130	-
11	1.088	1.050	*	1.131	1.110	-
12	1.069	1.030	*	1.085	1.060	+
13	1.038	1.000	**	1.037	1.014	*
14	1.043	1.010	*	1.029	1.006	*
15	1.054	1.020	*	1.026	1.004	*
16	1.058	1.020	*	1.022	1.000	**
17	1.074	1.030	*	1.030	1.008	*
18	1.083	1.040	*	1.037	1.015	*
19	1.120	1.080	+	1.040	1.018	*
20	1.200	1.160	-	1.069	1.050	+
21	1.240	1.190	-	1.090	1.070	+
22	1.320	1.270		1.130	1.110	-
23	1.340	1.290		1.168	1.144	-
24	1.344	1.290		1.200	1.170	-

Notes: 1. P_o = 50 psi.

2. The trigger in each harmonic is an internal pressure of 10^{-7} psi.

3. Each analysis included the indicated harmonic plus n=0.

4. Value assignments are according to the following legend:

SYMBOL	VALUE
**	absolutely necessary
*	great
+	moderate
-	slight
	none

TABLE VI

RELATIVE IMPERFECTION ROTATION COEFFICIENT SIZES, SHELL A-14

HARMONIC	Φ_s				Φ_θ			
	COS	VALUE	SIN	VALUE	COS	VALUE	SIN	VALUE
0	5.32	*	0.00		0.01		0.00	
1	3.18	*	2.03	*	0.34		0.40	
2	4.85	*	9.08	*	9.46	*	13.26	*
3	1.50	+	7.34	*	13.46	*	3.11	+
4	2.80	*	1.74	+	6.28	+	28.54	*
5	1.71	+	0.79		7.84	+	9.98	*
6	1.33	+	1.48	+	5.20	+	4.07	+
7	0.56		1.85	+	11.20	*	6.26	+
8	0.57		0.86	-	7.88	+	1.92	-
9	0.46		0.69		3.75	-	2.88	-
10	1.10	+	0.85	-	5.60	+	5.99	-
11	0.77		0.91	-	2.17	-	1.68	-
12	1.39	+	1.67	+	5.73	+	6.77	+
13	1.05	+	1.01	+	2.98	-	3.38	+
14	0.77		0.83	-	2.60	-	1.55	+
15	0.91	-	0.46		1.05	-	1.26	-
16	1.09	+	0.69		0.73		2.17	-
17	0.54		1.15	+	1.31	-	1.17	-
18	0.83	-	1.03	+	1.16	-	1.46	-
19	0.94	-	0.78		0.48		1.04	-
20	0.41		0.81	-	0.49		0.48	
21	0.42		0.68		0.45		0.19	
22	0.43		0.63		0.31		0.15	
23	0.68		0.58		0.11		0.10	
24	0.32		0.01		0.01		0.01	

Notes: 1. Numbers represent the sums of the absolute values of the imperfection rotation coefficients over the entire shell.

2. Value assignments were made according to the legend of Table V.

TABLE VII

HARMONIC SELECTION, SHELL A-14

HARMONIC	$\bar{\Phi}_s$ COEFFICIENT		$\bar{\Phi}_\theta$ COEFFICIENT		SHELL RESPONSE		SELECTED HARMONICS		
	COS	SIN	COS	SIN	FIXED SUPPORT	SIMPLE SUPPORT	FIXED SUPPORT		SIMPLE SUPPORT
							PRIMARY	ALTERNATE	PRIMARY
0	*						X	X	X
1	*	*					X	X	X
2	*	*	*	*			X	X	X
3	+	*	*	+			X	X	X
4	*	+	+	*			X	X	X
5	+		+	*			X	X	X
6	+	+	+	+					
7		+	*	+			X	X	X
8		-	+	-					
9			-	-	-				
10	+	-	+	+	*	-	X		
11		-	-	-	*	-	X		
12	+	+	+	+	*	+	X	X	
13	+	+	-	+	**	*	X	X	X
14		-	-	-	*	*	X	X	X
15	-		-	-	*	*	X	X	X
16	+			-	*	**		X	X
17		+	-	-	*	*		X	X
18	-	+	-	-	*	*			X
19	-			-	+	*			
20		-			-	+			
21					-	+			
22						-			
23						-			
24						-			

Notes: 1. Harmonic zero is required by SATANS-II.

2. Value assignments are taken from Tables V and VI.

3. 'X' indicates a selected harmonic.

coefficients (obtained from the periodogram) must be loaded into the SATANS-II arrays PHIXB and PHITB (performed in subroutine IMPERF) in the following order:

1. The first KMAX locations contain the $\bar{\phi}_s$'s (or $\bar{\phi}_\theta$'s) of mode 0.

2. The second KMAX locations contain the $\bar{\phi}_s$'s (or $\bar{\phi}_\theta$'s) of mode -i.

3. The third KMAX locations contain the $\bar{\phi}_s$'s (or $\bar{\phi}_\theta$'s) of mode +i.

4. The fourth KMAX locations contain the $\bar{\phi}_s$'s (or $\bar{\phi}_\theta$'s) of mode -j.

5. Et cetera, until all modes are entered.

KMAX is the FORTRAN name for the number of meridional stations to be included in the analysis. One special case must be considered where the shell has an initial or a final pole. In this case, rotations are zero in all but the n=1 harmonic. If this harmonic is included in the analysis, the imperfection rotation coefficients at the first (or last) station should all be set to zero except for the coefficients in the n=1 harmonic. In the case of an initial pole, the coefficient associated with the -1 index should be placed in the first location of the second set of KMAX locations in arrays PHIXB and PHITB, and the coefficient associated with the +1 index should be placed in the first location of the third set of KMAX locations. For a final pole, the coefficient associated with the -1 index should be placed in the first location of the next to last set of KMAX

locations, and the coefficient associated with the +1 index should be placed in the first location of the last set of KMAX locations.

To illustrate the effect of the selection of harmonics on the imperfections, equation (30) was solved using only the 13 selected harmonics. The resulting circumferential imperfection rotation representation at station 10 appears as curve B in Figure 18. To determine the effect of the inability to include all nine harmonics deemed important to the shell response with a fixed support, an alternate selection of harmonics was made for this case (see Table VII). A complete SATANS-II analysis was performed with this alternate selection and the results were found to differ from those using the primary harmonic selection by approximately 2%.

3. Results

Six analyses (boundary conditions A, B and C with clamped and simple supports) were made for each shell where the imperfections were reduced by a factor of 1000. These 'minute' imperfections acted only as triggers for the modal deflections and the resulting solutions gave the shell's consistent critical bifurcation buckling loads. These results are presented in Table VIII under the category of minute imperfections. Selected cases were analyzed where the minute imperfection triggers were replaced by a 10^{-7}psi internal pressure trigger and the results were identical.

TABLE VIII

BUCKLING LOADS, MINUTE VERSUS FULL IMPERFECTIONS

SUPPORT	DEGREE OF IMPERFECTION	BOUNDARY CONDITION	SHELL A-14		SHELL A-13	
			BUCKLING LOAD (lb/in)	REDUCTION (%)	BUCKLING LOAD (lb/in)	REDUCTION (%)
CLAMPED	Minute	A	47.20		35.60	
		B	51.90		50.60	
		C	56.00		55.10	
	Full	A	43.00	8.90	*	-
		B	46.60	10.20	40.70	19.55
		C	49.00	12.50	40.15	22.70
SIMPLE	Minute	A	48.00		46.60	
		B	51.10		50.60	
		C	54.10		55.30	
	Full	A	42.80	10.8	34.30	26.30
		B	46.10	9.8	38.00	24.90
		C	47.90	11.5	42.7	22.80

* insufficient data.

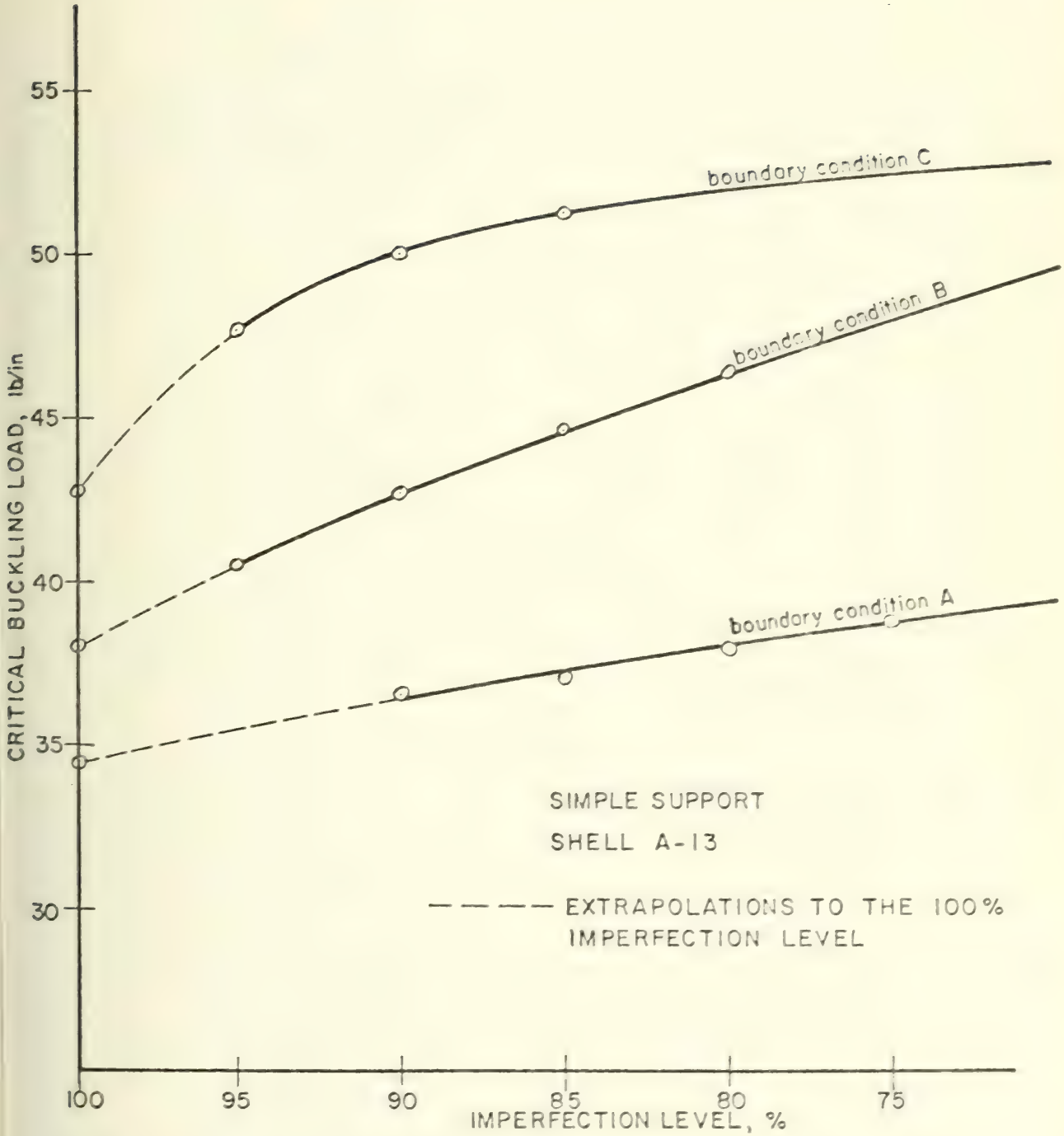
The same six analyses for each shell were also made using the full imperfections. Results for the final load at which a converged solution could be obtained are also shown in Table VIII along with the percent reduction in the buckling load for each case.

Since the full imperfection is introduced at the first load step as a pseudo load, the program requires considerably more iterations than normal to obtain the first

solution. The subsequent solution usually proceeds in a completely normal fashion. However, convergence difficulties may be encountered in cases where the imperfection level is high. This was the case with shell A-13, as some of its imperfection deviations exceeded 3½ shell wall thicknesses, and fluctuated over 6 thicknesses in a θ arc of only 53^{o}. For shell A-13, a 70% initial imperfection level was the maximum for which a converged solution could be obtained on the first load step. Consequently, buckling loads were computed with a maximum of 80%, 85%, 90% and 95% of the full imperfection data. For the 80% imperfection, 70% was introduced at load step one and 10% was added at load step two. For the three other solutions, 5% increments were added at various load steps. Since SATANS uses the total value of the variables and not the incremental values, each intermediate solution is correct for its given load and imperfection level, but the load-deflection characteristics are invalid. An imperfection level of 95% was the absolute maximum for which a converged solution could be obtained for for shell A-13, and a short 5% extrapolation was made to obtain the data presented in Table VIII for this shell.[5] This procedure is illustrated for the shell with a simple support in Figure 19. A comparison of load steps and iterations for both shells under these conditions is presented in Table IX.

---

[5]For shell A-13 with clamped boundary condition A, a converged solution was not obtained for an imperfection level above 85%. This condition is therefore not included in the results. A similar problem was encountered by Dantone /Ref. 5/ with this boundary condition.



CRITICAL BUCKLING LOAD vs IMPERFECTION LEVEL

FIGURE 19

TABLE IX

LOAD STEPS VERSUS ITERATIONS

SHELL A-14			SHELL A-13		
LOAD STEP	INCREMENTAL IMPERFECTION ADDITION	ITERATIONS REQUIRED	LOAD STEP	INCREMENTAL IMPERFECTION ADDITION	ITERATIONS REQUIRED
1	100%	7	1	70%	24
2		2	2	10%	6
3		2	3	5%	16
4		3	4		12
5		3	5		9
6		3	6		3
7		3	7		8
8		3	8		6
9		3	9	5%	19
10		3	10		3
11		3	11	5%	60
12		4	12		58
13		4	13		18
14		4	14		16
15		5	15		16
16		7	16		12
17		10			
18		16	1st load step size reduction		
1st load step size reduction			17		4
19		3	2nd load step size reduction		
20		2			
			18		3
2nd load step size reduction			19		2
21		3	3rd load step size reduction		
3rd load step size reduction			termination		
22		2	(P_{cr}=40.6 lb/in)		
termination					
(P_{cr}=46.0 lb/in)					

Note: This data corresponds to boundary condition B, simple support.

D. DISCUSSION

The SATANS-II investigation of the buckling load of the perfect cylinder (discussed in section V.C.1) indicates that a numerical analysis using 35 stations results in a higher predicted buckling load than the correct analytical buckling load. Consequently, the SATANS-II results for the prediction of the critical buckling load of the imperfect cylinder are expected to be high. Table X compares the results of the SATANS-II analysis (column A) with the actual experimental results of Ref. 6 (column E) and shows this to be the case. The use of more stations to lower the predicted buckling load can be accomplished only by either decreasing the number of harmonics considered in the analysis or by increasing overall program size. Both of these choices are impractical at the present time since the number of harmonics already in the analysis appears to be low considering the number required for shell response and the imperfection data, and the present program core storage requirement is large.

It is possible, however, to apply the SATANS-II results for the <u>percent reduction</u> in the buckling load due to the imperfections (see Table VIII and Column B, Table X) to a more accurate buckling load of the perfect shell from an analysis like that of Almroth to get a more accurate estimate of the buckling load of the imperfect shell. When the percent reduction from the consistent minute imperfection buckling load to the consistent full imperfection buckling

load, determined by SATANS-II, is applied to the accurate consistent bifurcation buckling load of Almroth (column C, Table X), the result is a prediction of the actual buckling load of the imperfect shell. These imperfect shell predicted buckling loads are shown in column D of Table X and are seen to be very close to the actual buckling loads of the imperfect shells, given in Column E of Table X.

The percent reductions in buckling loads determined by SATANS-II are considered to be applicable to the accurate consistent bifurcation buckling loads of Almroth since the imperfection characteristics are seen to be concentrated in the lower ten harmonics (see Table VI), where shell response does not appear to be largely affected by the number of meridional stations in the analysis (see Figures 14 and 15).

E. PROPOSED ANALYSIS PROCEDURE

When the SATANS-II predicted buckling load of the perfect shell is higher than the analytical prediction, as is the case with the extremely thin-walled cylinder, the following procedure is proposed.

1. Determine the consistent bifurcation buckling load as close as possible.

2. Determine the buckling load with SATANS-II using minute imperfections.

3. Determine the buckling load with SATANS-II using full imperfections and compute the percent reduction from step 2.

4. Apply this reduction to the buckling load of step 1.

TABLE X

PREDICTED BUCKLING LOADS OF THE IMPERFECT SHELL

	BOUNDARY CONDITION	A SATANS-II PREDICTIONS WITH FULL IMPERFECTIONS (lb/in)	B SATANS-II PERCENT REDUCTION (%)	C ALMROTH'S ANALYTICAL CONSISTENT PREDICTION [Ref. 14] (lb/in)	D PERCENT REDUCTION APPLIED TO ANALYTICAL PREDICTION (lb/in)	E ACTUAL EXPERIMENTAL RESULTS [Ref. 6] (lb/in)
SHELL A-14 CLAMPED	A	43.00	8.9	42.5	38.70	35.4
	C	49.00	12.5	42.0	36.75	
SHELL A-13 CLAMPED	C	40.15	27.2	42.0	30.50	31.4

This procedure is expected to work only when the result of step 2 is reasonably close to that of step 1.

VI. CONCLUSIONS

A. SATANS-I

The computer program SATANS-I has been constructed from SATANS. It is a more user-oriented version of the program by Ball, whose primary modification is the introduction of an equipment-independent graphical output ability.

B. SATANS-II

SATANS-II is a further modification of SATANS where the restriction that all applied loading and initial conditions be symmetrical about some meridional plane has been removed. In addition, SATANS-II possesses the capability of including experimentally-determined naturally-occurring imperfections or malformations induced during fabrication, mathematically described imperfections of any type, frequency or location, or graphically described imperfections due to damage suffered after fabrication. The imperfections are introduced into SATANS-II by using the imperfection deviations along discrete circumferences to determine the associated meridional and circumferential imperfection rotations at the mesh points, performing a Fourier expansion on these rotations and then use of the Fourier coefficients. In cases where the important shell response harmonics and the harmonics required for imperfection representation exceed the limits of SATANS-II, a selection must be made.

Instructions for the use of both SATANS-I and SATANS-II are contained in Appendix C and complete listings of both programs are given in other appendices. SATANS-I requires 210 k-bytes (53k words) of core on an IBM-360/67 computer using 'single precision,' while SATANS-II requires 426 k-bytes (107k words) of core. Imperfect shell analyses performed by SATANS-II in this report used a 35 station meridian and 25 circumferential modes. Execution times were an average of 60 minutes, although execution times of between 15 and 30 minutes can be expected for analyses using 15 or less modes.

C. IMPERFECT SHELLS

SATANS-II has demonstrated an ability to perform a numerical structural analysis on the arbitrarily imperfect shell of revolution. Imperfections of over three shell wall thicknesses in depth have been handled. Increasing solution difficulty relates directly to increasing imperfection complexity and increasing shell instability. In those cases where extremely thin-walled shells result in perfect (imperfection-free) buckling loads from SATANS that are higher than analytically determined buckling loads, a procedure has been devised where the percent reduction due to the introduction of full imperfections can be applied to the analytically determined buckling load, and an imperfect shell buckling load predicted. These predictions have been shown to be very close to the actual imperfect shell buckling loads for the two specific axially loaded cylinders considered.

ADDENDUM

A DIGITAL COMPUTER STUDY OF THE EFFECTS OF VARIOUS ARTIFICIAL BOUNDARY CONDITIONS AT THE POLE OF A SHALLOW SPHERICAL SHELL

I. INTRODUCTION

The design of many shell of revolution structures requires continuity of the structure at one or both poles. At such a pole, the radius becomes zero and special conditions must be imposed on the deflections and the stress resultants to assure finite strains and equilibrium. As a consequence of the difficulty of incorporating these conditions into a structural analysis computer program, many researchers model such a structure as having a small hole or a small rigid plug at the pole. This addendum investigates the applicability of the various artificial pole conditions by examining their effect on the stability of a clamped spherical cap subjected to axisymmetric and asymmetric pressure loads.

Various polar artificialities are discussed and the boundary equations associated with their use are given, as are those for the continuous pole. A determination of the applicability of the various poles is made by comparisons of their effect on the shell buckling loads and by comparing the errors in the normal displacement and meridional membrane force distributions incurred by their use.

The computer program SATANS, developed by Ball [Ref. 3], was used to conduct this study and selected results of the SATANS analyses are checked for validity with the computer programs BOSOR [Ref. 15] and STAGS [Ref. 15]. SATANS has a special routine that incorporates the proper conditions to be enforced at a pole. BOSOR also has this special ability but STAGS is not yet so equipped.[6]

SATANS, a one-dimensional finite difference equilibrium formulation based on Sanders' equations, was run on an IBM-360/67 computer under OS/MVT. BOSOR, a one-dimensional finite difference energy formulation based on the Donnell equations, was run on a UNIVAC-1108 computer under EXEC-VIII, while STAGS, a two-dimensional finite difference energy formulation was run on a CDC-6600 computer under COMPASS-12.

---

[6]As of February, 1972.

II. STATEMENT OF THE PROBLEM

A. GEOMETRY AND LOADING

The structure investigated is the clamped shallow spherical cap whose geometry can be specified by the single nondimensional parameter λ, where

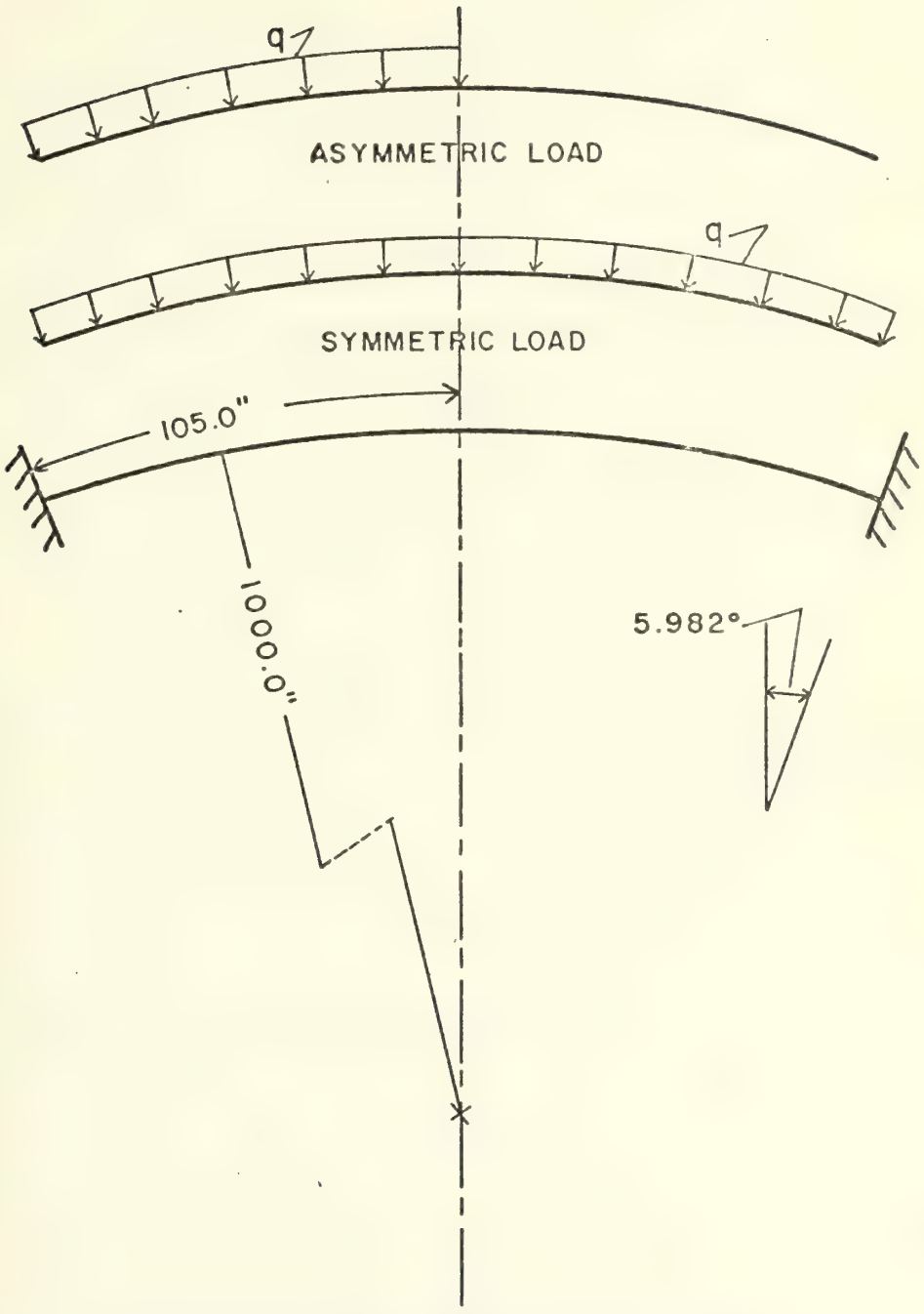
$$\lambda = 2\left(3(1-\nu^2)\right)^{\frac{1}{4}}\left(H/h\right)^{\frac{1}{2}} \qquad \text{(AD-1)}$$

and where H is the rise of the shell and h is its thickness. For this study, Poisson's ratio was taken as 0.3, Young's modulus as 30×10^6 psi and h as 1.0 inches. Shells of $\lambda = 6$ and $\lambda = 18$ were investigated however only the results of the $\lambda = 6$ shell are given in detail.

The classical buckling pressure of a complete sphere is given as q_o, where

$$q_o = 2Eh^2/R_s/\left(3(1-\nu^2)\right)^{\frac{1}{2}} \qquad \text{(AD-2)}$$

and where R_s is the radius of curvature. For this study, two loading conditions were used. The cap was subjected to a uniform normal pressure (symmetrical load) and a uniform normal pressure over half the cap (asymmetrical load). These loadings are shown in Figure 20. In each case, the magnitude of the applied pressure is q and the buckling results are presented as the ratio q/q_o.



$\lambda = 6$ SHALLOW SPHERICAL CAP
GEOMETRY AND LOADING

FIGURE 20

B. POLE CONDITIONS

A shell is considered to have a pole if the structure is continuous through its axis of revolution. At a pole, the radius is zero and special conditions must be imposed on deflections u, v and w, and on the meridional moment m_s to assure finite strains and equilibrium. The computer program SATANS is equipped with the capability to impose these special boundary conditions. A full derivation of the required conditions is given in Appendix D of Ref. 3. The results of that derivation are presented below.

For the zeroeth Fourier harmonic (n=0):

$$\begin{aligned} &u = v = m_{s\theta} = t_{s\theta} = 0 \\ &w' = m_s' = 0 \\ &m_{s\theta} = m_s \\ &t_\theta = t_s = b(1-\nu)\left(u'+\omega_s w+\tfrac{1}{2}\beta_s\right)-t_T \end{aligned} \qquad \text{(AD-3)}$$

For the first Fourier harmonic (n=1):

$$\begin{aligned} &(u\pm v) = w = u' = 0 \\ &m_s = m_\theta = m_{s\theta} = 0 \\ &t_s = t_\theta = t_{s\theta} = 0 \end{aligned} \qquad \text{(AD-4)}$$

When a double sign (±) appears, the upper sign applies for an initial pole and the lower sign applies for a final pole.

For the second harmonic (n=2):

$$\begin{aligned} u &= v = w = m_s' = 0 \\ t_\theta &= -t_s = \pm t_{s\theta} \\ m_\theta &= -m_s = \pm m_{s\theta} \\ t_s &= b(1-\nu)\left(u' + \tfrac{1}{2}\beta_s\right) \end{aligned} \qquad \text{(AD-5)}$$

For all higher harmonics (n>2):

$$\begin{aligned} u &= v = w = 0 \\ m_s &= m_\theta = m_{s\theta} = 0 \\ t_s &= t_\theta = t_{s\theta} = 0 \end{aligned} \qquad \text{(AD-6)}$$

For rotations at the pole, for the first harmonic (n=1):

$$\begin{aligned} \phi_s &= -w' + \omega_s u \\ \phi_\theta &= \mp \phi_s \\ \phi &= \tfrac{1}{2}\left(2v' \pm u'\right) \end{aligned} \qquad \text{(AD-7)}$$

and for all other harmonics,

$$\begin{aligned} \phi_s &= \phi_\theta = 0 \\ \phi &= \tfrac{1}{2}(2v' \pm nu') \end{aligned} \qquad \text{(AD-8)}$$

Numerous artificial boundary conditions can be substituted for the correct conditions at the pole. For example,

a small hole can be placed at the pole and can exist with the boundary completely free, or have conditions imposed such that the meridional rotation ϕ_s and deflection u are restrained. A meridional line load N_s of magnitude sufficient to replace that 'lost' due to the hole can be applied. Another artificial pole condition is that best described as a small rigid plug. Here, both the meridional as well as the circumferential deflection is restrained along with the meridional rotation. A normal line load -P whose total equals that loading 'lost' due to the hole or plug can be applied at the boundary of the hole or plug.

Preliminary numerical studies showed that the line loadings effected a noticeable change in the solution only in the immediate area of the boundary, hence, the line loads were not considered further. Preliminary studies also showed that the cap behavior exhibited only minor changes when the small hole with a partially restrained edge was substituted for that with a totally free edge. This investigation, therefore, concentrated on two artificial boundary conditions at a pole and on the continuous pole. The artificial pole conditions used and their associated mathematical expressions are:

1. The totally free hole:

$$N_s = 0 \; ; \; N_{s\theta} = 0 \; ; \; Q_s = 0 \; ; \; M_s = 0 \qquad \text{(AD-9)}$$

2. The rigid plug:

$$U = 0 \; ; \; V = 0 \; ; \; Q_s = 0 \; ; \; \phi_s = 0 \qquad \text{(AD-10)}$$

C. TYPES OF ANALYSIS

Three types of buckling analyses were conducted on the cap under each of the three pole conditions and these were:

1. The nonlinear axisymmetric analysis of the symmetrically loaded clamped spherical cap.

2. The bifurcation buckling analysis of the symmetrically loaded clamped spherical cap with nonlinear prebuckling axisymmetric displacements.

3. The nonlinear collapse analysis of the asymmetrically loaded clamped spherical cap.

In the artificial pole cases, hole and plug radii were taken as 0.17%, 1.70% and 3.30% of the shell meridian. The SATANS analyses used a 100 station meridian for the axisymmetric cases, a 66 station meridian with three harmonics for the bifurcation buckling cases and a 40 station meridian with five harmonics for the nonsymmetric collapse cases.

III. DISCUSSION

The results of the SATANS analyses are presented in Table XI for all cases. Also presented are the normal deflections of a point located at 47.5% of the meridian which is near the maximum displacement for each case.

A. THE $\lambda = 6$ SHALLOW CAP

1. The Axisymmetric Analysis

In Table XI, the buckling load is seen to be basically unaffected by the altered pole conditions. The normal displacement, however, is seen to show a marked sensitivity to the pole conditions. Figure 21 shows this sensitivity. In the figure, the upper curve is the deflection shape of the cap at incipient buckling with a continuous pole, and the lower curves represent the percent error induced in the normal displacement at incipient buckling by the introduction of artificial poles. The 0.17% plug and the 0.17% hole behaved identically. Holes, in general, produced an average 20% error in the normal displacement and this error existed over the entire shell. The plugs, preserving the continuity of the structure, produced an average 10% error, but again, this error extended over the entire shell. Figure 22 shows the behavior of the meridional force N_s for the continuous pole case, and the N_s error due to the artificial pole conditions, also at incipient buckling. Large errors existed in all cases over the first third of

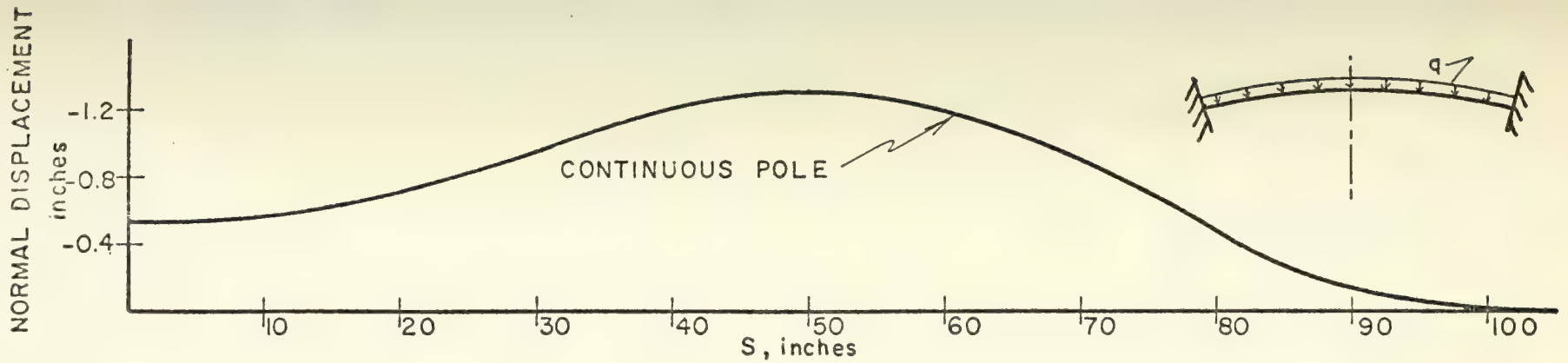
TABLE XI

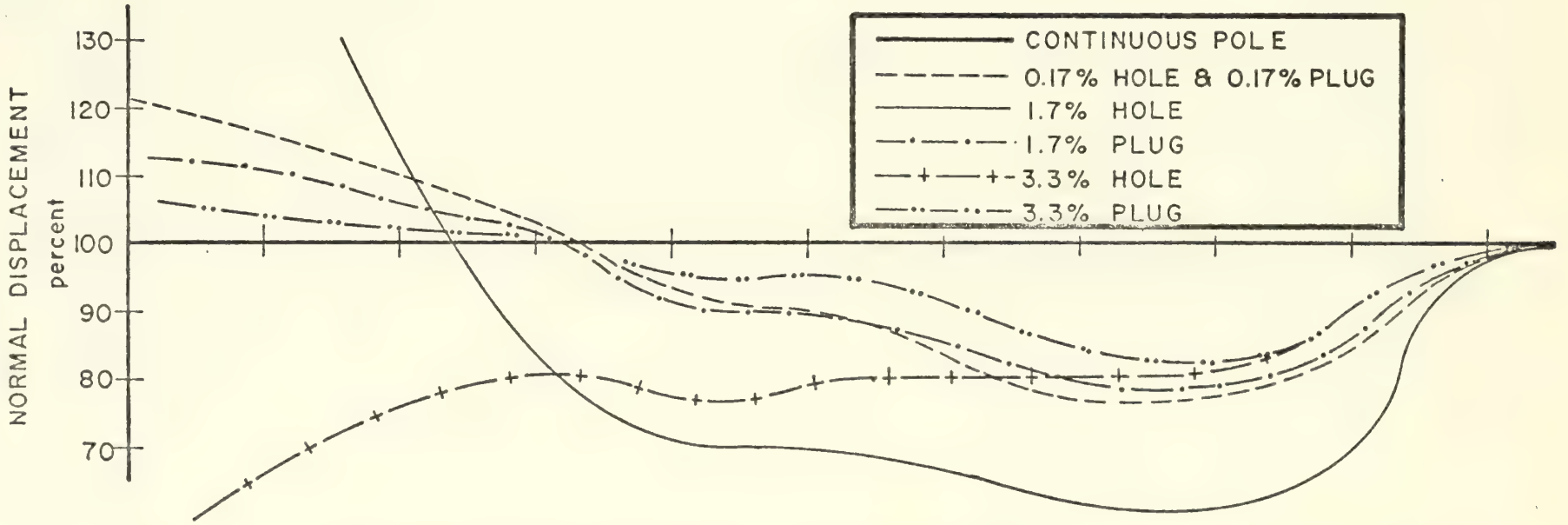
SHALLOW CAP (λ=6) BUCKLING LOADS AND DISPLACEMENTS AT A POINT

ANALYSIS TYPE	POLE CONDITION	MERIDIONAL STATIONS/ HARMONICS	q/q_o	NORMAL DEFLECTION AT 47.5% MERIDIAN (inches)
Axisymmetric Buckling - Symmetric Load	Pole	-	0.9950*	-
		30/1	1.0610	1.3720
		100/1	1.0200	1.3200
	-0.17% Hole-1.70% -3.30%	100/1	0.9720	1.1600
		100/1	0.9690	1.0560
		100/1	0.9620	1.0290
	-0.17% Plug-1.70% -3.30%	100/1	0.9720	1.2570
		100/1	0.9770	1.1630
		100/1	0.9900	1.2290
Bifurcation Buckling - Symmetric Load	Pole	-	0.7750*	-
		30/3	0.7851	0.6386
		66/3	0.7752	0.6188
	-0.17% Hole-1.70% -3.30%	66/3	0.7719	0.6356
		66/3	0.7835	0.6562
		66/3	0.7392	0.6264
	-0.17% Plug-1.70% -3.30%	66/3	0.7719	0.6356
		66/3	0.7726	0.6438
		66/3	0.7713	0.6513
Collapse Analysis - Asymmetric Load	Pole	-	0.7100†	-
		40/5	0.5391	0.9830
	-0.17% Hole-1.70% -3.30%	40/5	0.5363	1.0780
		40/5	0.5389	1.0060
		40/5	0.5962	1.4110
	-0.17% Plug-1.70% -3.30%	40/5	0.5978	0.9478
		40/5	0.6290	0.8100
		40/5	0.7576	0.7248

* Huang [Ref. 16]

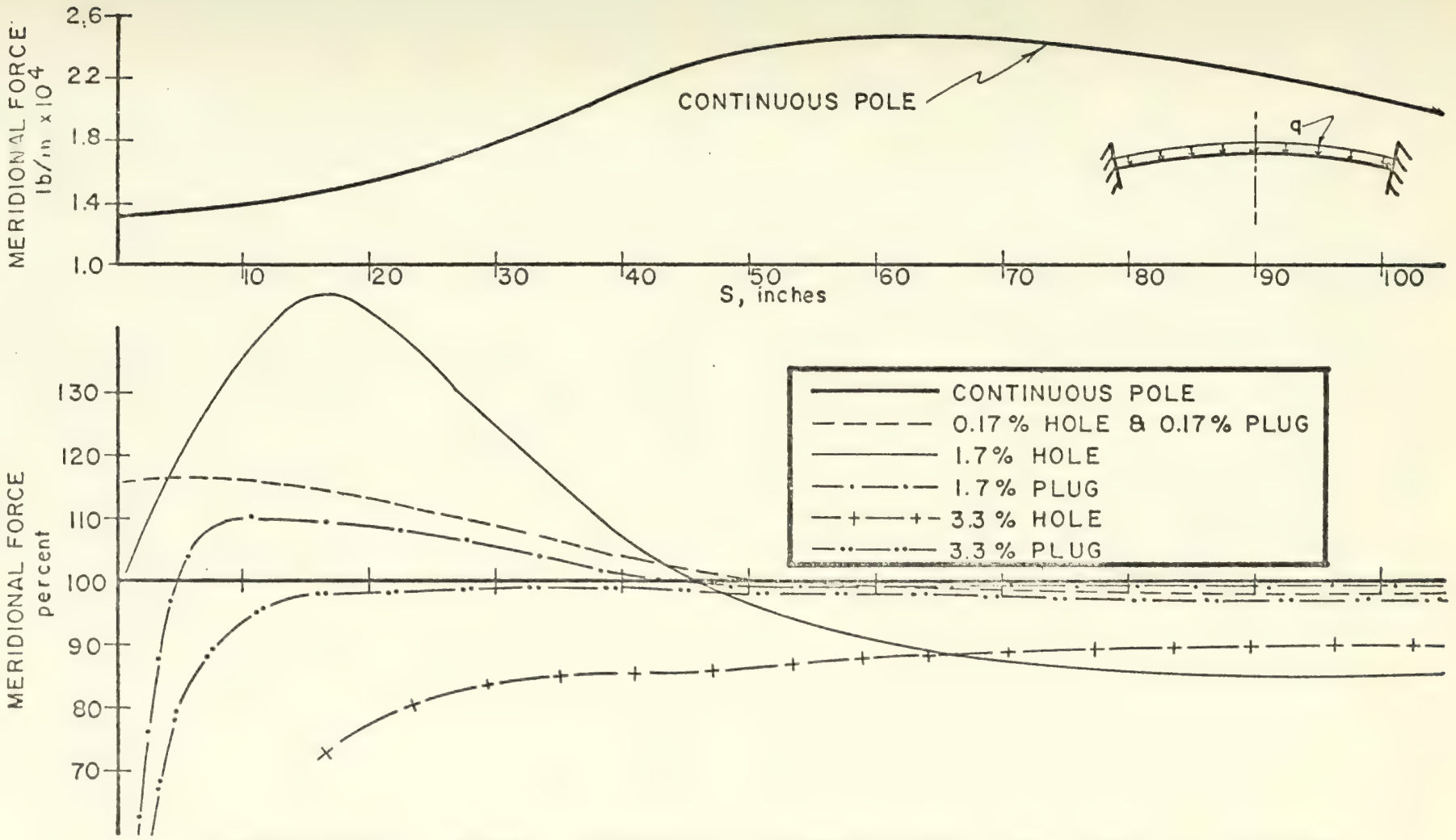
† Famili and Archer [Ref. 17]





NORMAL DISPLACEMENT ERROR, AXISYMMETRIC ANALYSIS, $\lambda = 6$ CAP

FIGURE 21



MERIDIONAL FORCE ERROR, AXISYMMETRIC ANALYSIS, $\lambda = 6$ CAP

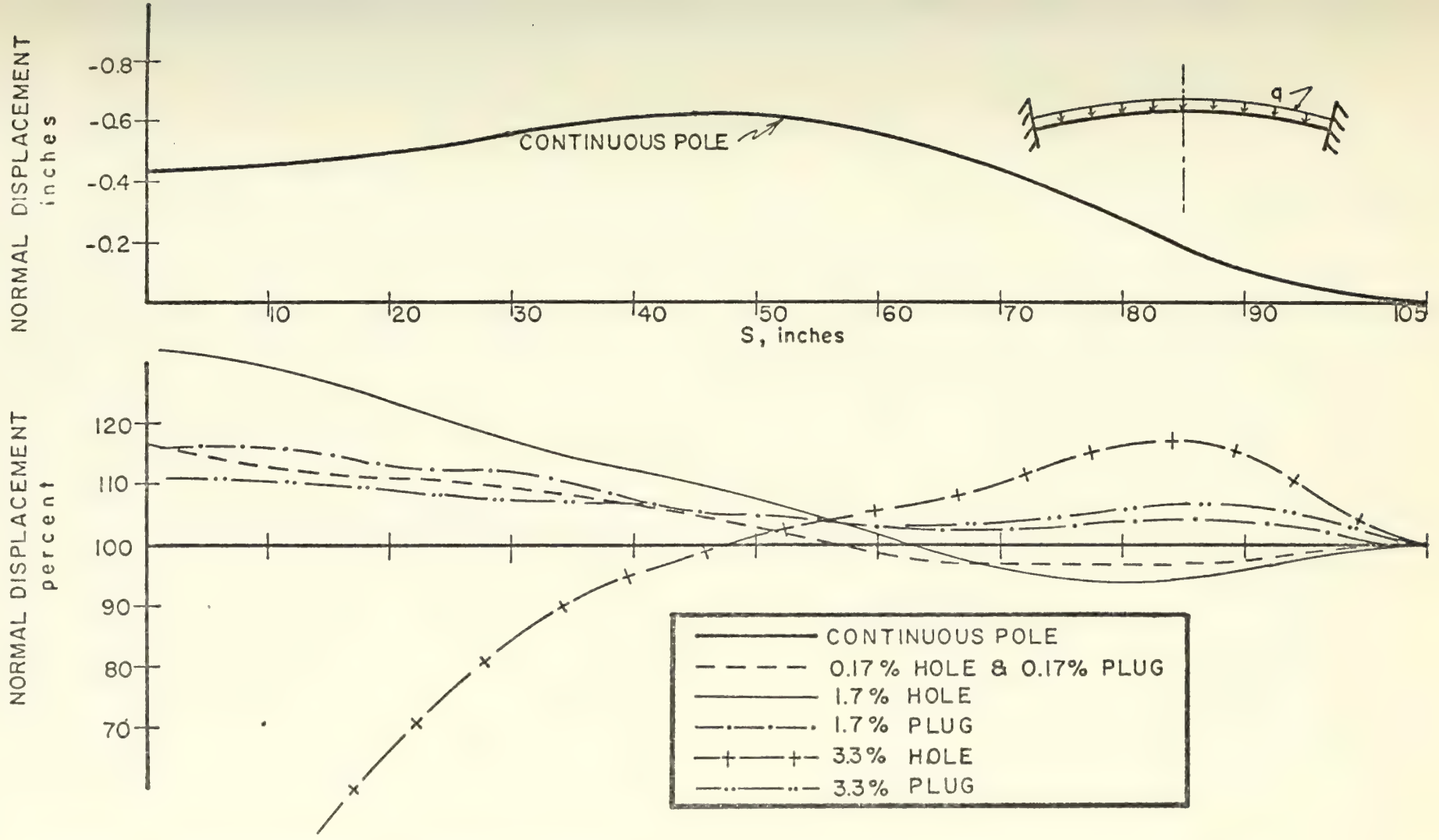
FIGURE 22

the meridian and were an average of 10% or less for the remainder of the meridian.

2. Bifurcation Buckling Analysis

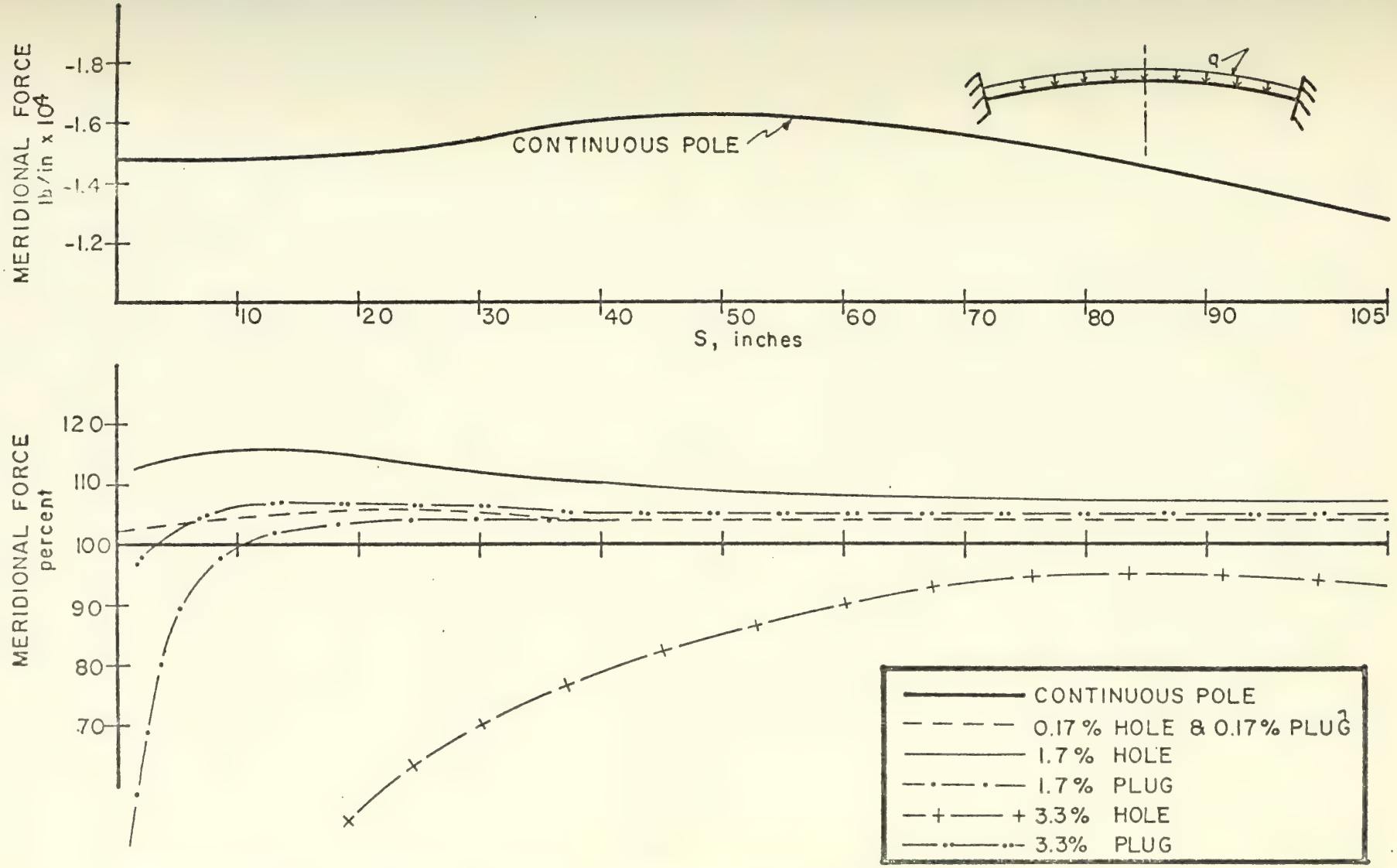
In this analysis, the shell buckling load appears to be insensitive to artificial pole conditions for the case of the plugs, and shows some sensitivity in the case of the holes. This is surprising since bifurcation buckling of the $\lambda = 6$ cap occurs in the $\cos(2\theta)$ mode. While equations (AD-8) show that no rotations exist at the pole in this mode, rotations certainly do exist at 1.7% and 3.3% meridian due to the buckling mode. The equations for the rigid plug's boundary conditions (equations (AD-10)) specifically prevent this rotation, while those for the hole (equations (AD-9)) do not.

Examination of Figures 23 and 24 show that the displacement and force distributions of the plugs at incipient buckling to be less erroneous, again, than the poles. In Figure 23, an average 10% displacement error occurs over the entire shell for the plugs, while holes of 1.7% and 3.3% meridian cause large errors over much of the shell. This trend is repeated in Figure 24, although the errors are not as large. From this analysis, the indications are that the 1.7% hole is a poor artificial pole and a 3.3% hole is unacceptable as it causes a 4.6% error in buckling load and over 40% error in some regions in both normal displacement and meridional force distributions. Larger hole sizes would apparently increase these errors.



NORMAL DISPLACEMENT ERROR, BIFURCATION BUCKLING ANALYSIS, $\lambda = 6$ CAP

FIGURE 23



MERIDIONAL FORCE ERROR, BIFURCATION BUCKLING ANALYSIS, $\lambda = 6$ CAP

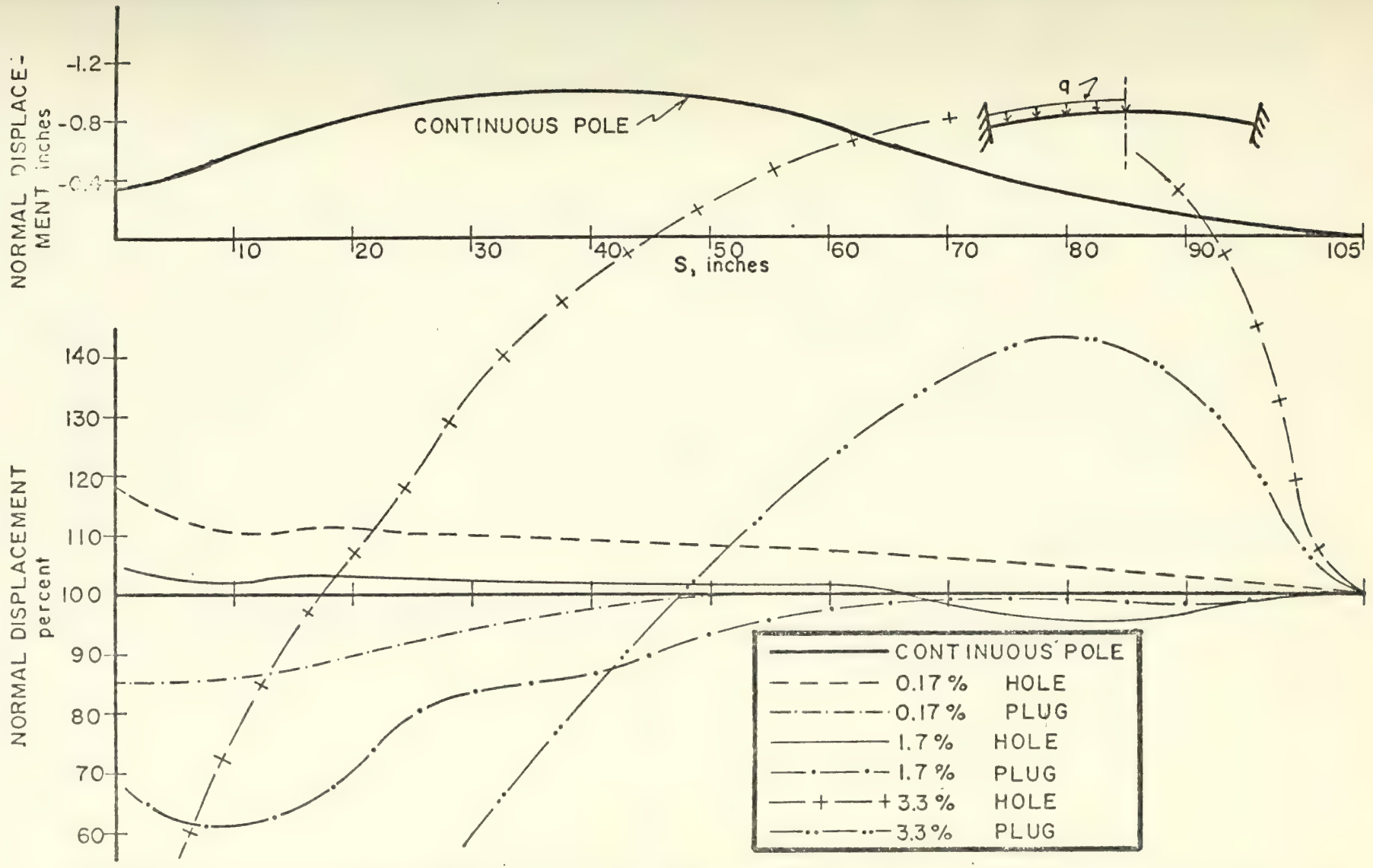
FIGURE 24

3. Nonsymmetric Collapse Analysis

The results for the collapse analysis are also presented in Table XI, and in Figures 25 and 26 for the $\theta = 0^{o}$ meridian, which is symmetrically located under the loaded region. In Table XI, it is shown that the buckling loads of the cap with rigid plugs are quite erroneous (10.8%, 16.6% and 41.4% for the 0.17%, 1.7% and 3.3% plugs, respectively) while those for the small holes are quite reasonable. In Figures 25 and 26, it is seen that the small holes cause an average error of 10% in both the normal displacement and meridional force distributions at incipient buckling over the entire shell, while the errors caused by the rigid plugs is large.

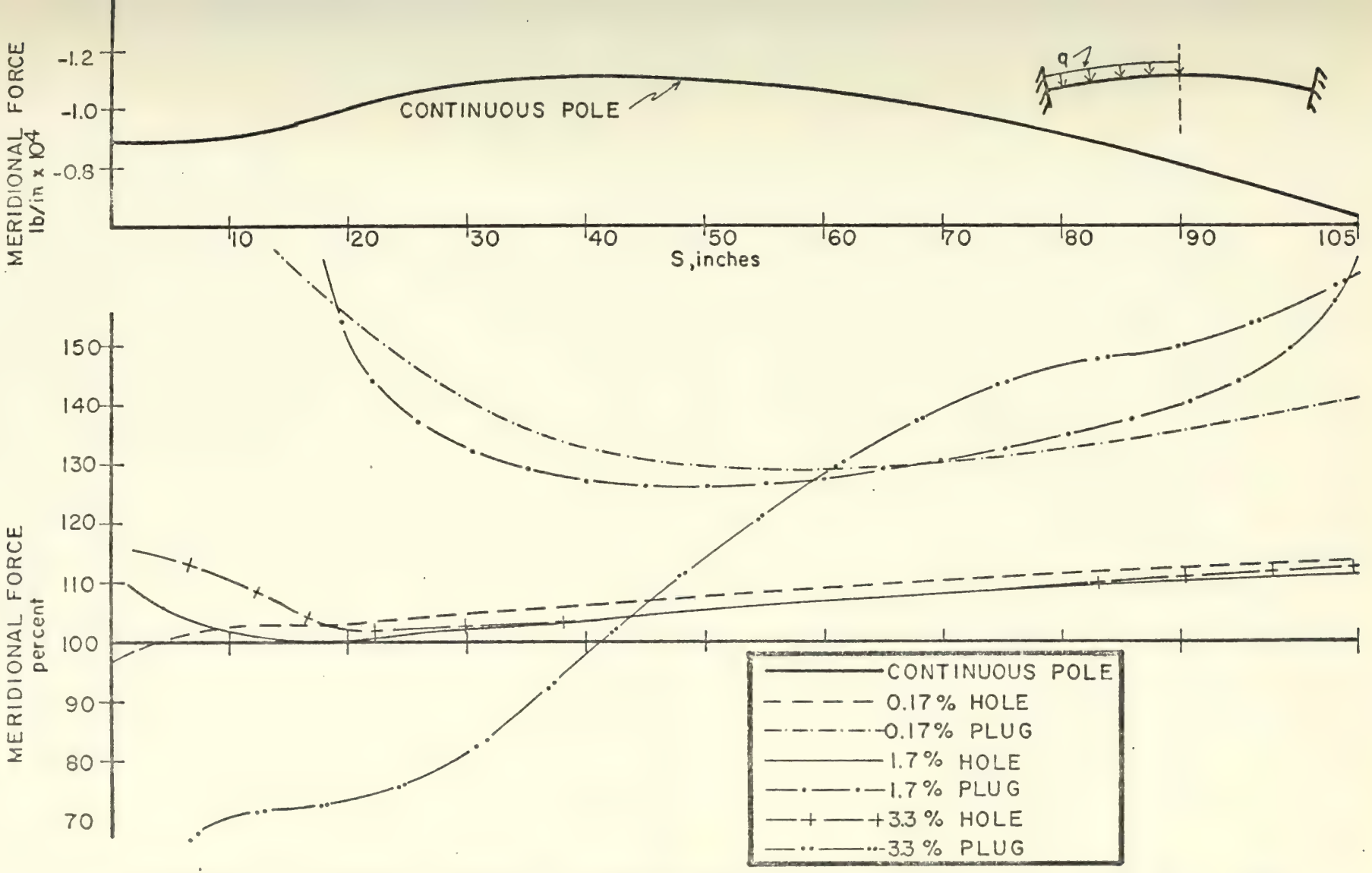
In an asymmetrically loaded shell, large rotations will exist at the pole and the restrictions imposed by the rigid plug appear to make it an unacceptable artificial pole for use in this case.

Figures 25 and 26 also indicate that a 1.7% hole is the best artificial pole for this case. Assuming such a 'best' hole size exists for another given shell or loading condition, an investigation similar to that presented here would have to be performed to find it. This is an inherently expensive and time consuming process and there is no guarantee that there is a 'best' hole size without its radius becoming so small that mathematical difficulties arise during the solution process. Consequently, the use of



NORMAL DISPLACEMENT ERROR, NONSYMMETRIC COLLAPSE ANALYSIS, λ=6 CAP

FIGURE 25



MERIDIONAL FORCE ERROR, NONSYMMETRIC COLLAPSE ANALYSIS, $\lambda = 6$ CAP

FIGURE 26

artificial pole conditions in nonsymmetric problems must be approached with caution.

B. THE $\lambda = 18$ SHALLOW CAP

A deeper shell with $\lambda = 18$ was investigated in a similar fashion and produced similar results. The shell buckling load was negligibly affected (except in those cases already mentioned for the $\lambda = 6$ shell) but displacement errors of up to 30% and meridional membrane force errors of an average 10-15% were noted. In general, the $\lambda = 18$ shell appeared less sensitive to the type of artificial pole (except in the asymmetric case), but as sensitive to any artificial pole as the shallower $\lambda = 6$ shell.

C. COMPARISONS WITH OTHER RESEARCHERS

As indicated in the introduction, various SATANS results were checked for validity with BOSOR and STAGS. These comparisons are presented with others in Table XII and very favorable agreement is seen to exist.

TABLE XII

SHALLOW CAP (λ=6) BUCKLING LOADS COMPARISON

ANALYSIS TYPE	POLE CONDITION	SOURCE	q/q_o	REMARKS
Axisymmetric Buckling	Pole	Huang	0.9950	Classical /Ref. 16/
		SATANS	1.0200	100 Stations
Bifurcation Buckling	Pole	Huang	0.7750	50 Stations /Ref. 16/
		BOSOR	0.7756	40 Stations /Ref. 15/
		SATANS	0.7752	66 Stations, 3 harmonics
	0.17% Hole	STAGS	0.7689	30x36 variably spaced grid /Ref. 15/
		BOSOR	0.7736	40 Stations
		SATANS	0.7719	66 Stations, 3 harmonics
	1.70% Plug	STAGS	0.7711	30x36 variable spaced grid
		BOSOR	0.7747	40 Stations
		SATANS	0,7716	66 Stations, 3 harmonics
Nonsymmetric Collapse	Pole	Famili & Archer	0.7100	7 Stations /Ref. 17/
		SATANS	0.6730	7 Stations, 4 harmonics /Ref. 18/*
		SATANS	0.6560	7 Stations, 10 harmonics *
		SATANS	0.5380	20 Stations, 4 harmonics *
		SATANS	0.5380	20 Stations, 10 harmonics *
		SATANS	0.5391	40 Stations, 5 harmonics
	1.70% Hole	STAGS	0.5233	22x92 variable spaced grid
		SATANS	0.5389	40 Stations, 5 harmonics

* This result and the following three were taken from a discussion of the effects of the number of stations and harmonics on the buckling loads of the clamped spherical cap, contained in Ref. 18.

IV. CONCLUSIONS

The stability of the symmetrically loaded clamped shallow spherical shell appears to be fairly insensitive to artificial pole conditions in the axisymmetric and bifurcation buckling cases if the radius of the artificiality is kept at or below 3% of the meridian. In the case of the asymmetrically loaded shell, this insensitivity only applies when small holes are substituted for the pole. In general, the following additional conclusions can be drawn.

1. Sizable errors are introduced into the normal displacement and meridional membrane force distributions by the introduction of an artificial pole.

2. The displacement and force error distributions are sensitive to the size and type of artificial pole and are usually greater than 10% with some cases of over 30%, and they exist over the entire shell meridian.

3. Small rigid plugs appear to be the best polar models for symmetrically loaded shells. Apparently their continuity-preserving characteristics greatly outweigh their rotational-restricting characteristics.

4. Small holes appear to be the best polar models for asymmetrically loaded shells. Here, the rotational restrictions of the rigid plug make its use unacceptable. For this type of loading, an investigation will be required to determine the optimum hole size.

5. In general, the greater the displacement, or the higher the membrane force that would exist at a pole, the greater the error introduced by the use of an artificial pole.

APPENDIX A

NONLINEAR TERMS

The nonlinear terms of equations (15) are given below. Terms involving the initial imperfection rotation terms along have been omitted as they cause no straining.

$$
\begin{aligned}
\Phi_s^2 = (\sigma_o/E_o)^2 \Bigg[& \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big)^2 + \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big)^2 \\
& + 2 \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \\
& + 2 \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_0^\infty \overline{\phi}_s^1(n) \operatorname{Cos} n\theta \Big) \\
& + 2 \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_1^\infty \overline{\phi}_s^2(n) \operatorname{Sin} n\theta \Big) \\
& + 2 \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_0^\infty \overline{\phi}_s^1(n) \operatorname{Cos} n\theta \Big) \\
& + 2 \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_1^\infty \overline{\phi}_s^2(n) \operatorname{Sin} n\theta \Big) \Bigg] \\
\equiv \sigma_o/E_o & \Big(\sum_0^\infty \beta_s^1(n) \operatorname{Cos} n\theta + \sum_1^\infty \beta_s^2(n) \operatorname{Sin} n\theta \Big)
\end{aligned}
\tag{A-1}
$$

$$\Phi_\theta^2 = (\sigma_o/E_o)^2\Big[\Big(\sum_1^\infty \phi_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)^2 + \Big(\sum_0^\infty \phi_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big)^2$$
$$+ 2\Big(\sum_1^\infty \phi_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)\Big(\sum_0^\infty \phi_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big)$$
$$+ 2\Big(\sum_1^\infty \phi_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)\Big(\sum_1^\infty \overline{\phi}_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)$$
$$+ 2\Big(\sum_1^\infty \phi_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)\Big(\sum_0^\infty \overline{\phi}_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big) \qquad \text{(A-2)}$$
$$+ 2\Big(\sum_0^\infty \phi_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big)\Big(\sum_1^\infty \overline{\phi}_\theta^1(n)\,\mathrm{Sin}\ n\theta\Big)$$
$$+ 2\Big(\sum_0^\infty \phi_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big)\Big(\sum_0^\infty \overline{\phi}_\theta^2(n)\,\mathrm{Cos}\ n\theta\Big)\Big]$$
$$\equiv \sigma_o/E_o\ \Big(\sum_0^\infty \beta_\theta^1(n)\,\mathrm{Cos}\ n\theta + \sum_1^\infty \beta_\theta^2(n)\,\mathrm{Sin}\ n\theta\Big)$$

$$\Phi^2 = (\sigma_o/E_o)^2\Big[\Big(\sum_1^\infty \phi^1(n)\,\mathrm{Sin}\ n\theta\Big)^2 + \Big(\sum_0^\infty \phi^2(n)\,\mathrm{Cos}\ n\theta\Big)^2$$
$$+ 2\Big(\sum_1^\infty \phi^1(n)\,\mathrm{Sin}\ n\theta\Big)\Big(\sum_0^\infty \phi^2(n)\,\mathrm{Cos}\ n\theta\Big)\Big] \qquad \text{(A-3)}$$
$$\equiv \sigma_o/E_o\ \Big(\sum_1^\infty \beta^1(n)\,\mathrm{Sin}\ n\theta + \sum_0^\infty \beta^2(n)\,\mathrm{Cos}\ n\theta\Big)$$

$$
\begin{aligned}
\Phi_s\Phi_\theta = (\sigma_o/E_o)^2 \Big[& \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \Big) \\
& + \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \Big) \\
& + \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_1^\infty \overline{\phi}_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_0^\infty \phi_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_0^\infty \overline{\phi}_\theta^2(n) \operatorname{Cos} n\theta \Big) \\
& + \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_1^\infty \overline{\phi}_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_1^\infty \phi_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_0^\infty \overline{\phi}_\theta^2(n) \operatorname{Cos} n\theta \Big) \\
& + \Big(\sum_0^\infty \overline{\phi}_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_0^\infty \overline{\phi}_s^1(n) \operatorname{Cos} n\theta \Big) \Big(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \Big) \\
& + \Big(\sum_1^\infty \overline{\phi}_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_1^\infty \phi_\theta^1(n) \operatorname{Sin} n\theta \Big) \\
& + \Big(\sum_1^\infty \overline{\phi}_s^2(n) \operatorname{Sin} n\theta \Big) \Big(\sum_0^\infty \phi_\theta^2(n) \operatorname{Cos} n\theta \Big) \Big] \\
\equiv \sigma_o/E_o & \Big(\sum_0^\infty \beta_{s\theta}^1(n) \operatorname{Cos} n\theta + \sum_1^\infty \beta_{s\theta}^2(n) \operatorname{Sin} n\theta \Big)
\end{aligned}
\qquad \text{(A-4)}
$$

APPENDIX B

EIGENVALUE PROBLEM MATRIX ELEMENT DEFINITIONS

The elements of the $/\bar{A}/$ and $/\bar{B}/$ matrices used in the eigenvalue solution are defined below.

$$
\begin{aligned}
A_{11} &= G_{11} - E_{11}Q^2 \\
A_{12} &= F_{12}Q \\
A_{13} &= F_{13}Q \\
A_{21} &= -F_{21}Q \\
A_{22} &= -E_{22}Q^2 + G_{22} - G_{24}G_{42}/G_{44} \\
A_{23} &= G_{23} + E_{23}Q^2 - G_{24}(G_{43} - E_{43}Q^2)/G_{44} \\
A_{31} &= -F_{31}Q \\
A_{32} &= G_{32} - E_{32}Q^2 - G_{42}(G_{34} - E_{34}Q^2)/G_{44} \\
A_{33} &= G_{33} - E_{33}Q^2 - (G_{34} - E_{34}Q^2)(G_{43} - E_{43}Q^2)/G_{44}
\end{aligned}
\tag{B-1}
$$

where

$$Q = m\pi/L$$

and where

$$
\begin{aligned}
B_{11} &= n^2/4 \\
B_{12} &= nQ/4 \\
B_{13} &= 0 \\
B_{21} &= n^2/4 \\
B_{22} &= Q^2/4 \\
B_{23} &= 0 \\
B_{31} &= 0 \\
B_{32} &= 0 \\
B_{33} &= Q^2
\end{aligned}
\tag{B-2}
$$

and where all elements of the 4 x 4 matrices $\underline{\overline{E}}$, $\underline{\overline{F}}$ and $\underline{\overline{G}}$ are zero except for the following, defined as

$$
\begin{aligned}
E_{11} &= b \\
E_{22} &= \tfrac{1}{2}(1-\nu)b + 9(1-\nu)d/(8R_\theta^2) \\
E_{23} &= 3(1-\nu)dn/(2R_\theta^2) \\
E_{32} &= E_{23} \\
E_{33} &= 2(1-\nu)dn^2/R_\theta^2 \\
E_{34} &= 1 \\
E_{43} &= -d
\end{aligned}
\qquad \text{(B-3)}
$$

$$
\begin{aligned}
F_{12} &= \tfrac{1}{2}(1-\nu)bn/R_\theta - 3(1-\nu)dn/(8R_\theta^3) \\
F_{13} &= b\nu/R_\theta - \tfrac{1}{2}(1-\nu)n^2/R_\theta^3 \\
F_{21} &= -F_{12} \\
F_{31} &= -F_{13}
\end{aligned}
\qquad \text{(B-4)}
$$

$$
\begin{aligned}
G_{11} &= -\tfrac{1}{2}(1-\nu)bn^2/R_\theta^2 - (1-\nu)dn^2/(8R_\theta^4) \\
G_{22} &= -bn^2/R_\theta^2 - (1-\nu)(1+\nu)dn^2/R_\theta^4 \\
G_{23} &= -bn/R_\theta^2 - (1-\nu)(1+\nu)dn^3/R_\theta^4 \\
G_{24} &= -\nu n/R_\theta^2 \\
G_{32} &= G_{23} \\
G_{33} &= -\nu n^2/R_\theta^2 \\
G_{42} &= \nu dn/R_\theta^2 \\
G_{43} &= \nu dn^2/R_\theta^2 \\
G_{44} &= -1
\end{aligned}
\qquad \text{(B-5)}
$$

and where

$$
\begin{aligned}
b &= \{(1-\nu)(1+\nu)\}^{-1} \\
d &= (b/12)(h/r)^2
\end{aligned}
\qquad \text{(B-6)}
$$

APPENDIX C

INPUT DATA GUIDE FOR SATANS-I AND SATANS-II

Card	Columns	Format	Item	Example	Meaning
1	1-72	18A4	TITLE	-	Enter any 72 characters
2	1- 5	I5	NO	1	The problem number, 0<NO<10000.
2	6-10	L5	$DYNAMC	F	For a static analysis, set $DYNAMC = F.
				T	For a dynamic analysis, set $DYNAMC = T.
2	11-15	I5	IMODE	0	For no modal output data.
				1	For modal output data.
2	16-20	I5	NDIMEN	0	Dimensional output data.
				1	Nondimensional output.
2	21-25	I5	NTHMAX	8	Summed solution will be printed at NTHMAX meridians, 0<=NTHMAX<=36.
2	26-30	I5	IFREQ	3	Solution will be printed at the first station, every subsequent IFREQ station and the last station, 0<IFREQ<=KMAX.
2	31-35	I5	IPRINT	2	Every IPRINT converged solution will be printed.
2	36-40	I5	IBCINL	-1	If the shell has a pole at the first station.
				0	If the shell has no pole at the first station.
2	41-45	I5	IBCFNL	-1	If the shell has a pole at the last station.
				0	If the shell has no pole at the last station.

Card	Columns	Format	Item	Example	Meaning
2	46-50	I5	KMAX	35	Number of meridional stations. Note: KMAX<201 for SATANS-I without plots and KMAX<101 for SATANS-I with plots or for SATANS-II.
2	51-55	I5	MNMAX	7	Number of series coefficients used to describe the initial conditions, pressure and thermal loads (and initial imperfections if using SATANS-II). Note: For SATANS-I: 0<MAXM<11 and KMAX*MAXM<201. For SATANS-II: 0<MAXM<26 and KMAX*MAXM<951.
2	61-65	I5	LSMAX	1	For a linear analysis.
				99	Use many load steps for a nonlinear static analysis.
				750	For a dynamic analysis, LSMAX is the number of time increments, where LSMAX=$T_{max}/\Delta T$.
2	66-70	I5	LCHMAX	3	The number of load step size reductions in a static analysis, recommended range = 2-4.
				0	For a dymanic analysis.
2	71-75	I5	ITRMAX	1	For a linear analysis.
				30	The number of iterations at a load or time step. For a nonlinear analysis, suggested range = 10-30, up to 50 for special cases.
2	76-80	I5	IC	0	Initial conditions. Set to 0 for a static analysis, or for a dynamic analysis where the shell is at rest at T=0.
				1	For a dynamic analysis with initial conditions.

Card	Columns	Format	Item	Example	Meaning
3	1-12	E12.3	NU	0.3	Poisson's ratio, ν.
3	12-24	E12.3	SIGO	1000.0	Reference stress level, σ_o.
				1.0	If the input data is dimensional.
3	24-36	E12.3	ELAST	.3E8	Reference modulus of elasticity, E_o.
				1.0	If the input data is dimensional.
3	37-48	E12.3	TKN	.4E-2	Reference thickness, h_o.
				1.0	If the input data is dimensional.
3	49-60	E12.3	CHAR	8.16	Characteristic shell dimension, a.
				1.0	If the input data is dimensional.
3	61-72	E12.3	TEEO	0.0	If a static analysis.
				.996E-5	Reference time, T_o.
4	1-12	E12.3	DELOAD	0.2	For a static analysis, DELOAD is the load increment. It remains unchanged until the solution fails to converge in ITRMAX iterations, when it is reduced by a factor of five. A maximum of LCHMAX such reductions will occur.
				.1823E-6	For a dynamic analysis, DELOAD is the nondimensional time increment.
4	13-24	E12.3	EPS	0.01	The convergence criterion recommended range of 0.01<EPS<0.001.

Card 4A is only included for a SATANS-II analysis.

Card	Columns	Format	Item	Example	Meaning
4A	1- 5	I5	JUMP	1	For an analysis using single series expansions.
				2	For an analysis using double series expansions.

Card	Column	Format	Item	Example	Meaning
4A	5-10	I5	MPERFS	0	An analysis without imperfections.
				1	An analysis with imperfections. Note: if JUMP=2, MPERFS may be 0 or 1. if JUMP=1, MPERFS must be 0 if MPERFS=1, JUMP must be 2.

Include as many cards 5 as necessary to specify NTHMAX meridians. If NTHMAX equals 0, omit card 5.

5	1-72	6E12.3		10.0	A list of circumferential coordinates θ, in degrees and tenths, where the solution printout is desired. The list must have NTHMAX entries.

If IBCINL=-1, omit cards 6 through 14. If IBCFNL=-1, omit cards 15 through 23. Cards 6 through 23 describe the boundary conditions at the first, and then at the last station. The boundary conditions exist on the total variables, not on the individual harmonics. Loadings applied through specification of boundary conditions are taken in the zeroeth harmonic (n=0) only, as the column matrix $\{\ell\}$ is set to zero for harmonics greater than zero. The boundary conditions are dimensional. The format of cards 6 through 23 is 4E16.8.

Card 6,15	Card 7,16	Card 8,17	Card 9, 18		
$\Omega(1,1)$	$\Omega(1,2)$	$\Omega(1,3)$	$\Omega(1,4)$	N_s	
$\Omega(2,1)$	$\Omega(2,2)$	$\Omega(2,3)$	$\Omega(2,4)$	$N_{s\theta}$	+
$\Omega(3,1)$	$\Omega(3,2)$	$\Omega(3,3)$	$\Omega(3,4)$	Q_s	
$\Omega(4,1)$	$\Omega(4,2)$	$\Omega(4,3)$	$\Omega(4,4)$	Φ_s	

Card 10,19	Card 11,20	Card 12,21	Card 13,22		Card 14,23
$\Lambda(1,1)$	$\Lambda(1,2)$	$\Lambda(1,3)$	$\Lambda(1,4)$	U	$\ell(1)$
$\Lambda(2,1)$	$\Lambda(2,2)$	$\Lambda(2,3)$	$\Lambda(2,4)$	V	$\ell(2)$
$\Lambda(3,1)$	$\Lambda(3,2)$	$\Lambda(3,3)$	$\Lambda(3,4)$	W	$\ell(3)$
$\Lambda(4,1)$	$\Lambda(4,2)$	$\Lambda(4,3)$	$\Lambda(4,4)$	M_s	$\ell(4)$

Card 24 is:
1. Included for a SATANS-I static analysis.
2. Included but blank for a SATANS-I dynamic analysis.
3. Omitted for a SATANS-II analysis.

Card	Column	Format	Item	Example	Meaning
24	1- 2	L2	$PLOTS	F	Indicates plots are NOT desired.
				T	Indicates plots are desired.
24	3- 4	L2	$MODAL	F	Indicates plots are for summed solutions only
				T	Indicates plots are for modal solutions only.

For the remainder of card 24 entries, 0 indicates that no plots are desired for the particular item, and 1 indicates that they are desired. All graphs are plotted as the indicated item versus the station number. If a complete plot is desired, insure IFREQ=1.

24	5- 6	I2	IRADII	1	Plot the radii as computed by subroutine GEOM.
24	7- 8	I2	IGAMMA	1	Plot ρ'/ρ as computed by subroutine GEOM.
24	9-10	I2	IOMEGS	1	Plot ω_s as computed by subroutine GEOM.
24	11-12	I2	IOMEGT	1	Plot ω_θ as computed by subroutine GEOM.
24	13-14	I2	IDEOMS	1	Plot ω_s' as computed by subroutine GEOM.
24	15-16	I2	IBSTIF	1	Plot the stiffness d as computed by subroutine BDB.

Card	Column	Format	Item	Example	Meaning
24	17-18	I2	IDSTIF	1	Plot the stiffness d as computed by the subroutine BDB.
24	19-20	I2	IBBSTF	1	Plot the stiffness db/dζ as computed by subroutine BDB.
24	21-22	I2	IDDSTF	1	Plot the stiffness dd/dζ as computed by subroutine BDB.
24	23-24	I2	IPR	1	Plot the normal component of the pressure load.
24	25-26	I2	IPS	1	Plot the meridional component of the pressure load.
24	27-28	I2	IPT	1	Plot the circumferential component of the pressure load.
24	29-30	I2	ITT	1	Plot the thermal load.
24	31-32	I2	IMT	1	Plot the thermal moment.
24	33-34	I2	IDTT	1	Plot d/dζ of the thermal load.
24	35-36	I2	IDMT	1	Plot d/dζ of the thermal moment.
24	37-38	I2	INS	1	Plot the meridional membrane force distribution.
24	39-40	I2	INTH	1	Plot the circumferential membrane force distribution
24	41-42	I2	INSTH	1	Plot the meridio-circumferential membrane force distribution.
24	43-44	I2	IQS	1	Plot the transverse force distribution.
24	45-46	I2	IMS	1	Plot the meridional moment distribution.
24	47-48	I2	IMTH	1	Plot the circumferential moment distribution.

Card	Column	Format	Item	Example	Meaning
24	49-50	I2	IMSTH	1	Plot the meridio-circumferential moment distribution.
24	51-52	I2	IU	1	Plot the meridional displacement distribution.
24	53-54	I2	IV	1	Plot the circumferential displacement distribution.
24	55-56	I2	IW	1	Plot the normal displacement distribution.
24	57-58	I2	IPHIS	1	Plot the meridional rotation distribution.
24	59-60	I2	IPHIT	1	Plot the circumferential rotation distribution.
24	61-62	I2	IPHI	1	Plot the meridio-circumferential rotation distribution.

Insert imperfection data here for a SATANS-II analysis with imperfections. Insure format of the imperfection data is compatible with that specified in the user-written subroutine IMPERF.

25	1- 2	I2	IRNAGN	0	Indicates this is the only run.
				1	Indicates another run is to be made. Add another complete set of data cards after this card is IRNAGN=1.

APPENDIX D

LISTING OF SATANS-I USER PREPARED SUBROUTINES

```
C*****************************************************************
C     THIS PROGRAM SERVES AS THE 'MAIN' PROGRAM, AND CALLS SUBROUTINE  *
C     'SATANS'.  AT THE END OF A RUN, IT CHECKS 'IRNAGN' TO SEE IF     *
C     THERE IS ANOTHER RUN.                                            *
C*****************************************************************
      COMMON /BLRUN/ IRNAGN
    1 CALL SATANS
      CALL FLYNVY
      IF (IRNAGN.EQ.1) GO TO 1
      STOP
      END




      SUBROUTINE GEOM
C*****************************************************************
C     THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL GEOMETRY FUNCTIONS   *
C     OF THE SHELL.                                                    *
C*****************************************************************
      REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
      COMMON /IBL4/ KMAX,KL
      COMMON /BL8/  R(200),GAM(200),OMT(200)
      COMMON /BL11/ OMXI(200),PHEE,TO,T2
      COMMON /BL17/ DEL
      COMMON /BL20/ DEOMX(200)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(200)
C*****************************************************************
C     GEOMETRY FOR CYLINDER A-14                                       *
C     R(S) = INFINITY                                                  *
C     R(TH)= 4.0 INCHES                                                *
C     S    = 8.16 INCHES                                               *
C     KMAX = 35 STATIONS                                               *
C     A    = 8.16 INCHES (CHARACTERISTIC SHELL DIMENSION)              *
C*****************************************************************
```

```
      DEL=8.16/(8.16*34.0)
      RO=4.0/8.16
      RORECP=1.0/RO
      DO 1 K=1,KMAX
      R(K)=RO
      GAM(K)=0.0
      OMT(K)=RORECP
      OMXI(K)=0.0
    1 DEOMX(K)=0.0
      RETURN
      END
```

```
      SUBROUTINE BDB(K,B,DB,D,DD)
C*****************************************************************
C     THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL INPLANE AND BENDING  *
C     STIFFNESSES OF THE SHELL.                                        *
C*****************************************************************
      REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),
     1              V3(10),W3(10)
      COMMON /BL17/ DEL
      COMMON /BL32/ TKN,ELAST,CHAR,SIG/
C*****************************************************************
C     NONDIMENSIONAL STIFFNESS QUANTITIES FOR CYLINDER A-14            *
C     WHERE:          E    = 15,800,000 PSI                            *
C                     E(0)= 15,800,000 PSI                             *
C                     H(0)= 0.00437 INCHES                             *
C                     A    = 8.16 INCHES                               *
C*****************************************************************
      B=1./.91
      D=B/12.
      DB=0.
      DD=0.
      RETURN
      END
```

```
      SUBROUTINE PLOAD(K)
C*****************************************************************
C     THIS SUBROUTINE ESTABLISHES THE NONDIMENSIONAL FOURIER      *
C     COEFFICIENTS OF THE LOADS APPLIED TO THE SHELL, AND ALSO    *
C     ESTABLISHES THE MODES TO BE CONSIDERED IN THE ANALYSIS.     *
C     ESTABLISHMENT OF MODES CAN ALSO BE ACCOMPLISHED IN SUBROUTINE*
C     TLOAD.                                                      *
C*****************************************************************
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ NN(10),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /INL8/ LSTEP,ITR
      COMMON /BL3/  PR(10),PX(10),PT(10)
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD
      COMMON /BL8/  R(200),GAM(200),OMT(200)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(200)
C*****************************************************************
C     THIS LOADING IS FOR ESTABLISHMENT OF THE CYLINDER A-14 BIFURCA-*
C     TION BUCKLING LOAD IN THE 21ST HARMONIC.                    *
C     THE PRESSURE LOADING DEFINED HERE ACTS AS MODAL TRIGGERS, AS THE
C     AXIAL LOAD IS INTRODUCED THROUGH THE BOUNDARY CONDITIONS.   *
C*****************************************************************
      A=CHAR/(SIGO*TKN)
      NN(1)=0
      NN(2)=21
      PR(1)=0.0000001*A
      PR(2)=0.0000001*A
      RETURN
      END
```

```
      SUBROUTINE TLOAD(K)
C*************************************************************
C     THIS SUBROUTINE DESCRIBES THE THERMAL LOADING ON THE SHELL.     *
C*************************************************************
      REAL NU
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ NN(10),MNINIT
      COMMON /IBL8/ LSTEP,ITR
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),
     1              V3(10),W3(10)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
C*************************************************************
C     THERE IS NO THERMAL LOADING ON CYLINDER A-14.                   *
C*************************************************************
      RETURN
      END




      SUBROUTINE INITL
C*************************************************************
C     THIS SUBROUTINE DESCRIBES THE INITIAL CONDITIONS FOR DYNAMIC RUNS*
C*************************************************************
      IMPLICIT LOGICAL*1 ($)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ NN(10),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL9/ MAXM
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL100/ TEEO,$DYNAMC
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD
      COMMON /BL104/ ZDOT(4,220)
C*************************************************************
C     THERE ARE NO INITIAL CONDITIONS FOR A STATIC ANALYSIS.          *
C*************************************************************
RETURN
      RETURN
      END
```

APPENDIX E

LISTING OF SATANS-I

THIS LISTING COMPRISES THE STORABLE PACKAGE OF SATANS-I AND CONTAINS THE FOLLOWING SUBROUTINES:

SATANS	STATIC	DYNAMC	OUTPUT
PLOT2	FLYNVY	FORCE	MATINV
ABC	PANDD	MODES	HJ
POLE	PLOTIT	SCALIT	DRAWIT
ROUND	XANDZ	PLOT1	PMATRX
UPDATE	INLPOL	FNLPOL	PHIBET
EFG	TEAETA		

```
      SUBROUTINE SATANS                                                      00000010
C*************************************************************************00000020
C     THIS SUBROUTINE READS ALL DATA, PRINTS THE OUTPUT TITLE PAGE,     *00000030
C     AND PASSES CONTROL TO ONE OF THE MAJOR CONTROLLING SUBROUTINES    *00000040
C*************************************************************************00000050
      IMPLICIT LOGICAL*1 ($)                                                 00000060
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                             00000070
      COMMON /IBL1/ MNMAX                                                    00000080
      COMMON /IBL4/ KMAX,KL                                                  00000090
      COMMON /IBL5/ IBCINL,IBCFNL                                            00000100
      COMMON /IBL9/ MAXM                                                     00000110
      COMMON /IBL10/ IFREQ,NTHMAX                                            00000120
      COMMON /IBL13/ ITRMAX,LSMAX                                            00000130
      COMMON /BL13/ OMEGL1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),            00000140
     1              UNIT(4,4)                                               00000150
      COMMON /BL15/ NJ,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),     00000160
     1              V3(10),W3(10)                                           00000170
      COMMON /BL16/ EPS                                                      00000180
      COMMON /BL18/ EL1(4),ELL(4)                                            00000190
      COMMON /BL19/ TH(36)                                                   00000200
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                      00000210
      COMMON /BL100/ TEEO,$DYNMC                                             00000220
      COMMON /BL102/ DELOAD                                                  00000230
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,      00000240
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,      00000250
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,         00000260
     3                IPHIT,IPHI,$PLOTS,$MODAL                              00000270
```

```
      CCMMON /BLDATA/ TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC                    00000280
      COMMON /BLRUN/ IRNAGN                                                     00000290
      DIMENSION TITLE (18)                                                      00000300
C*************************************************************************00000310
C     READ IN DATA FOR THIS RUN                                           *00000320
C*************************************************************************00000330
      READ (5,100) TITLE                                                        00000340
      READ (5,101) NO,$DYNMC,IMODE,NDIMEN,NTHMAX,IFREQ,IPRINT,IBCINL,           00000350
     1             IBCFNL,KMAX,MNMAX,MAXM,LSMAX,LCHMAX,ITRMAX,IC              00000360
      READ (5,102) NU,SIGO,ELAST,TKN,CHAR,TEEO                                  00000370
      READ (5,103) DELOAD,EPS                                                   00000380
      IF (NTHMAX.EQ.0) GO TO 1                                                  00000390
      READ (5,104) (TH(NTH),NTH=1,NTHMAX)                                       00000400
      DO 2 NTH=1,NTHMAX                                                         00000410
    2 TH(NTH)=TH(NTH)*6.2831853/360.0                                           00000420
    1 IF (IBCINL.EQ.0) READ (5,105) OMEG1,CAPL1,EL1                             00000430
      IF (IBCFNL.EQ.0) READ (5,105) OMEGL,CAPLL,ELL                             00000440
      READ (5,106) $PLOTS,$MODAL,IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,            00000450
     1             IBSTIF,IDSTIF,IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,      00000460
     2             IDMT,INS,INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,           00000470
     3             IPHIS,IPHIT,IPHI                                           00000480
      READ (5,107) IRNAGN                                                       00000490
C*************************************************************************00000500
C     THIS CONCLUDES DATA READ-IN FOR THIS RUN.                           *00000510
C     PRINT OUT PERTINENT DATA.                                           *00000520
C*************************************************************************00000530
      WRITE (6,200) NO                                                          00000540
      WRITE (6,201) TITLE                                                       00000550
      WRITE (6,202)                                                             00000560
      IF (IBCINL.LT.0) WRITE (6,203)                                            00000570
      IF (IBCINL.LT.0) GO TO 3                                                  00000580
      WRITE (6,204)                                                             00000590
      WRITE (6,205) (OMEG1(1,J),J=1,4),(CAPL1(1,J),J=1,4),EL1(1)                00000600
      WRITE (6,206) (OMEG1(2,J),J=1,4),(CAPL1(2,J),J=1,4),EL1(2)                00000610
      WRITE (6,207) (OMEG1(3,J),J=1,4),(CAPL1(3,J),J=1,4),EL1(3)                00000620
      WRITE (6,208) (OMEG1(4,J),J=1,4),(CAPL1(4,J),J=1,4),EL1(4)                00000630
    3 IF (IBCFNL.LT.0) WRITE (6,209)                                            00000640
      IF (IBCFNL.LT.0) GO TO 4                                                  00000650
      WRITE (6,210)                                                             00000660
      WRITE (6,205) (OMEGL(1,J),J=1,4),(CAPLL(1,J),J=1,4),ELL(1)                00000670
      WRITE (6,206) (OMEGL(2,J),J=1,4),(CAPLL(2,J),J=1,4),ELL(2)                00000680
      WRITE (6,207) (OMEGL(3,J),J=1,4),(CAPLL(3,J),J=1,4),ELL(3)                00000690
      WRITE (6,208) (OMEGL(4,J),J=1,4),(CAPLL(4,J),J=1,4),ELL(4)                00000700
    4 WRITE (6,211)                                                             00000710
      IF ($DYNMC) GO TO 5                                                       00000720
      WRITE (6,212) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,LCHMAX,EPS                    00000730
      WRITE (6,213) CHAR,TKN,ELAST,SIGO,NU                                      00000740
      IF (.NOT.$PLOTS) GO TO 6                                                  00000750
```

```
      ICHCK1=IABS(IRADII)+IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)                00000760
     1       +IABS(IDEOMS)+IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)              00000770
     2       +IABS(IDDSTF)+IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)             00000780
     3       +IABS(IMT)+IABS(IDTT)+IABS(IDMT)                                  00000790
      IF (ICHCK1.GT.0) WRITE (6,214)                                           00000800
      IF (ICHCK1.EQ.0) GO TO 8                                                 00000810
      IF (IRADII.NE.0) WRITE (6,215)                                           00000820
      IF (IGAMMA.NE.0) WRITE (6,216)                                           00000830
      IF (IOMEGS.NE.0) WRITE (6,217)                                           00000840
      IF (IOMEGT.NE.0) WRITE (6,218)                                           00000850
      IF (IDEOMS.NE.0) WRITE (6,219)                                           00000860
      IF (IBSTIF.NE.0) WRITE (6,220)                                           00000870
      IF (IDSTIF.NE.0) WRITE (6,221)                                           00000880
      IF (IBBSTF.NE.0) WRITE (6,222)                                           00000890
      IF (IDDSTF.NE.0) WRITE (6,223)                                           00000900
      IF (IPR   .NE.0) WRITE (6,224)                                           00000910
      IF (IPS   .NE.0) WRITE (6,225)                                           00000920
      IF (IPT   .NE.0) WRITE (6,226)                                           00000930
      IF (ITT   .NE.0) WRITE (6,227)                                           00000940
      IF (IMT   .NE.0) WRITE (6,228)                                           00000950
      IF (IDTT  .NE.0) WRITE (6,229)                                           00000960
      IF (IDMT  .NE.0) WRITE (6,230)                                           00000970
    8 ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)               00000980
     1       +IABS(IMTH)+IABS(IMSTH)+IABS(IU)+IABS(IV)+IABS(IW)                00000990
     2       +IABS(IPHIS)+IABS(IPHIT)+IABS(IPHI)                               00001000
      IF((ICHCK1.GT.0).AND.$MODAL) WRITE (6,231)                               00001010
      IF((ICHCK1.GT.0).AND..NOT.$MODAL) WRITE (6,232)                          00001020
      IF (ICHCK1.EQ.0) GO TO 9                                                 00001030
      IF (INS   .NE.0) WRITE (6,233)                                           00001040
      IF (INTH  .NE.0) WRITE (6,234)                                           00001050
      IF (INSTH.NE.0) WRITE (6,235)                                            00001060
      IF (IQS   .NE.0) WRITE (6,236)                                           00001070
      IF (IMS   .NE.0) WRITE (6,237)                                           00001080
      IF (IMTH  .NE.0) WRITE (6,238)                                           00001090
      IF (IMSTH.NE.0) WRITE (6,239)                                            00001100
      IF (IU    .NE.0) WRITE (6,240)                                           00001110
      IF (IV    .NE.0) WRITE (6,241)                                           00001120
      IF (IW    .NE.0) WRITE (6,242)                                           00001130
      IF (IPHIS.NE.0) WRITE (6,243)                                            00001140
      IF (IPHIT.NE.0) WRITE (6,244)                                            00001150
      IF (IPHI  .NE.0) WRITE (6,245)                                           00001160
    9 GO TO 6                                                                  00001170
    5 WRITE (6,246) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,EPS                          00001180
      WRITE (6,247) CHAR,TKN,ELAST,SIGO,TEEO,NU                                00001190
    6 IF (NTHMAX.EQ.0) GO TO 7                                                 00001200
      WRITE (6,248)                                                            00001210
      WRITE (6,249) (TH(NTH),NTH=1,NTHMAX)                                     00001220
    7 IF (NDIMEN.EQ.1) WRITE (6,250)                                           00001230
```

```
      IF (NDIMEN.EQ.0) WRITE (6,251)                                    00001240
C*****************************************************************00001250
C     CALL APPROPRIATE CONTROLLING SUBPROGRAM                          *00001260
C*****************************************************************00001270
      IF ($DYNMC) CALL DYNAMC                                           00001280
      IF (.NOT.$DYNMC) CALL STATIC                                      00001290
C*****************************************************************00001300
C     READ STATEMENT FORMATS FOLLOW                                    *00001310
C*****************************************************************00001320
  100 FORMAT (18A4)                                                     00001330
  101 FORMAT (I5,L5,14I5)                                               00001340
  102 FORMAT (6E12.4)                                                   00001350
  103 FORMAT (2E12.4)                                                   00001360
  104 FORMAT (6E12.4)                                                   00001370
  105 FORMAT (4E16.8)                                                   00001380
  106 FORMAT (2L2,29I2)                                                 00001390
  107 FORMAT (I2)                                                       00001400
C*****************************************************************00001410
C     WRITE STATEMENT FORMATS FOLLOW                                   *00001420
C*****************************************************************00001430
  200 FORMAT ('1',48X,'--PROBLEM NUMBER',I5,'--',///)                   00001440
  201 FORMAT (' ',38X,18A4,///)                                         00001450
  202 FORMAT (' ',49X,'--INPUT DATA RECORD--',//'     THE BOUNDARY CONDI00001460
     1TIONS ARE: ',//)                                                  00001470
  203 FORMAT (' ',T10,'THE SHELL HAS AN INITIAL POLE',/)                00001480
  204 FORMAT (' ',T10,'AT THE INITIAL EDGE',//2X,'---------------OMEGA B00001490
     1AR---------------',15X,'---------------LAMDA BAR---------------',100001500
     22X,'---EL-----',/)                                                00001510
  205 FORMAT (' ','(',4E10.3,')  NS         (',4E10.3,')  U        ',E1000001520
     1.3)                                                               00001530
  206 FORMAT (' ','(',4E10.3,')  NST   +    (',4E10.3,')  V    =   ',E1000001540
     1.3)                                                               00001550
  207 FORMAT (' ','(',4E10.3,')  QS         (',4E10.3,')  W        ',E1000001560
     1.3)                                                               00001570
  208 FORMAT (' ','(',4E10.3,')  PHIS       (',4E10.3,')  MS       ',E1000001580
     1.3)                                                               00001590
  209 FORMAT (' ',T10,'THE SHELL HAS A FINAL POLE',/)                   00001600
  210 FORMAT (' ',T10,'AT THE FINAL EDGE',//2X,'---------------OMEGA BAR00001610
     1---------------',15X,'---------------LAMDA BAR---------------',1200001620
     2X,'---EL-----',/)                                                 00001630
  211 FORMAT (////)                                                     00001640
  212 FORMAT (' ',4X,'NUMBER OF STATIONS----------------------',I3/5X,'NO00001650
     1UMBER OF MODES---------------------',I3/5X,'INCREMENTAL LOAD F00001660
     2ACTOR-----------------',F6.3/5X,'MAXIMUM NUMBER OF LOAD STEPS-----00001670
     3-------',I3/5X,'MAXIMUM NUMBER OF ITERATIONS------------',I3/5X,'M00001680
     4AXIMUM NUMBER OF LOAD FACTOR CHANGES---',I3/5X,'CONVERGENCE CRITER00001690
     5ION-------------------',F6.4//)                                   00001700
  213 FORMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION----------',E12.4/5X00001710
```

```
     1,'REFERENCE THICKNESS----------------------',E12.4/5X,'REFERENCE EL00001720
     2ASTICITY----------------------',E12.4/5X,'REFERENCE STRESS----------00001730
     3------------',E12.4/5X,'POISSONS RATIO--------------------------------00001740
     4',E12.4//)                                                           00001750
  214 FORMAT (' ',T10,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED GE00001760
     1OMETRY, STIFFNESS OR LOADING QUANTITIES:'/)                          00001770
  215 FORMAT ('+',T70,'RADIUS'/)                                           00001780
  216 FORMAT ('+',T70,'GAMMA'/)                                            00001790
  217 FORMAT ('+',T70,'OMEGA-S'/)                                          00001800
  218 FORMAT ('+',T70,'OMEGA-THETA'/)                                      00001810
  219 FORMAT ('+',T70,'DEOMEGA-S'/)                                        00001820
  220 FORMAT ('+',T70,'B-STIFFNESS'/)                                      00001830
  221 FORMAT ('+',T70,'D-STIFFNESS'/)                                      00001840
  222 FORMAT ('+',T70,'B-PRIME'/)                                          00001850
  223 FORMAT ('+',T70,'D-PRIME'/)                                          00001860
  224 FORMAT ('+',T70,'PRESSURE LOADING - NORMAL'/)                        00001870
  225 FORMAT ('+',T70,'PRESSURE LOADING - MERIDIONAL'/)                    00001880
  226 FORMAT ('+',T70,'PRESSURE LOADING - CIRCUMFERENTIAL'/)               00001890
  227 FORMAT ('+',T70,'THERMAL LOADING'/)                                  00001900
  228 FORMAT ('+',T70,'THERMAL MOMENT'/)                                   00001910
  229 FORMAT ('+',T70,'D-THERMAL LOADING'/)                                00001920
  230 FORMAT ('+',T70,'D-THERMAL MOMENT'/)                                 00001930
  231 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED MO00001940
     1DAL QUANTITIES:'/)                                                   00001950
  232 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED SU00001960
     1MMED QUANTITIES:'/)                                                  00001970
  233 FORMAT ('+',T70,'N-S'/)                                              00001980
  234 FORMAT ('+',T70,'N-THETA'/)                                          00001990
  235 FORMAT ('+',T70,'N-S-THETA'/)                                        00002000
  236 FORMAT ('+',T70,'Q-S'/)                                              00002010
  237 FORMAT ('+',T70,'M-S'/)                                              00002020
  238 FORMAT ('+',T70,'M-THETA'/)                                          00002030
  239 FORMAT ('+',T70,'M-S-THETA'/)                                        00002040
  240 FORMAT ('+',T70,'U'/)                                                00002050
  242 FORMAT ('+',T70,'W'/)                                                00002060
  241 FORMAT ('+',T70,'V'/)                                                00002070
  243 FORMAT ('+',T70,'PHI-S'/)                                            00002080
  244 FORMAT ('+',T70,'PHI-THETA'/)                                        00002090
  245 FORMAT ('+',T70,'PHI'/)                                              00002100
  246 FORMAT (' ',4X,'NUMBER OF STATIONS----------------------',I3/5X,'N00002110
     1UMBER OF MODES-------------------------',I3/5X,'INCREMENTAL TIME--00002120
     2----------------------',E9.4/5X,'MAXIMUM NUMBER OF TIME STEPS-----00002130
     3-------',I5/5X,'MAXIMUM NUMBER OF ITERATIONS------------',I3/5X,'C00002140
     4ONVERGENCE CRITERION--------------------',F6.4//)                   00002150
  247 FORMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION----------',E12.4/5X00002160
     1,'REFERENCE THICKNESS----------------------',E12.4/5X,'REFERENCE EL00002170
     2ASTICITY----------------------',E12.4/5X,'REFERENCE STRESS----------00002180
     3------------',E12.4/5X,'REFERENCE TIME--------------------------00002190
```

```
     4',E12.4/5X,'POISSONS RATIO--------------------',E12.4//)                    00002200
 248 FORMAT (' ',T20,'CIRCUMFERENTIAL COORDINATES FOR THE PRINT RECORD,00002210
    1 IN RADIAN MEASURE, ARE:'/)                                                  00002220
 249 FORMAT (' ',T10,6F16.10)                                                     00002230
 250 FORMAT (///' ',T20,'THE DATA PRINTED IS NON-DIMENSIONAL')                    00002240
 251 FORMAT (///' ',T20,'THE DATA PRINTED IS DIMENSIONAL')                        00002250
     RETURN                                                                       00002260
     END                                                                          00002270
```

```
      SUBROUTINE STATIC                                                         00000010
C*****************************************************************************00000020
C     THIS SUBROUTINE IS THE MAJOR CONTROLLING ROUTINE FOR ALL STATIC  *00000030
C     ANALYSIS PROBLEMS.  IT CONDUCTS INITIALIZATION, PREPARES INPUT   *00000040
C     MATERIAL FOR PLOTTING IF REQ'D, DETERMINES SHELL GEOMETRY, STIFF-*00000050
C     NESSES,LOADING (PHYSICAL AND/OR THERMAL), AND INITIAL CONTIDIONS.*00000060
C     IT CONTROLS PROBLEM SOLUTION PROCEDURE.                          *00000070
C*****************************************************************************00000080
      IMPLICIT LOGICAL*1 ($)                                                    00000090
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                                00000100
      COMMON /IBL1/ MNMAX                                                       00000110
      COMMON /IBL2/ N(10),MNINIT                                                00000120
C                   'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG              00000130
      COMMON /IBL3/ M0,M1,M2,M3                                                 00000140
      COMMON /IBL4/ KMAX,KL                                                     00000150
      COMMON /IBL5/ IBCINL,IBCFNL                                               00000160
      COMMON /IBL6/ KLL                                                         00000170
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),               00000180
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                       00000190
      COMMON /IBL8/ LSTEP,ITR                                                   00000200
      COMMON /IBL9/ MAXM                                                        00000210
      COMMON /IBL10/ IFREQ,NTHMAX                                               00000220
      COMMON /IBL11/ ICORFL,IPASS                                               00000230
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                          00000240
      COMMON /IBL13/ ITRMAX,LSMAX                                               00000250
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                      00000260
      COMMON /BL3/  PR(10),PX(10),PT(10)                                        00000270
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),              00000280
```

```
     1                ZF3M(4,4,10),ZF4M(4,4,10)                                      00000290
      COMMON /BL5/    TT(10),MT(10),DT(10),DMT(10)                                   00000300
C                     'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPOL & FNLPOL           00000310
      COMMON /BL6/    Z(4,220),SOE,OSE,ALOAD                                         00000320
      COMMON /BL7/    D1,S1                                                          00000330
      COMMON /BL8/    R(200),GAM(200),OMT(200)                                       00000340
      COMMON /BL9/    FFS(4,10),EL1S(4),GEES(4,10)                                   00000350
      COMMON /BL10/   PHIX(10),PHIT(10),PHI(10)                                      00000360
      COMMON /BL11/   OMXI(200),PHEE,T0,T2                                           00000370
      COMMON /BL12/   TDLI,TDEL                                                      00000380
      COMMON /BL13/   OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),                  00000390
     1                UNIT(4,4)                                                      00000400
      COMMON /BL14/   LAM2,LSD18,LSD1N                                               00000410
      COMMON /BL15/   NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),           00000420
     1                V3(10),W3(10)                                                  00000430
      COMMON /BL16/   EPS                                                            00000440
      COMMON /BL17/   DEL                                                            00000450
      COMMON /BL18/   EL1(4),ELL(4)                                                  00000460
      COMMON /BL19/   TH(36)                                                         00000470
      COMMON /BL20/   DEOMX(200)                                                     00000480
      COMMON /BL23/   JAY(4,4),H(4,4)                                                00000490
      COMMON /BL24/   DL(4,4,10),DG(4,4,10),DF(4,4,10)                               00000500
      COMMON /BL25/   E(4,4),F(4,4),G(4,4)                                           00000510
      COMMON /BL27/   BX3(10),BT3(10),BXT3(10),BE3(10)                               00000520
      COMMON /BL28/   EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)            00000530
      COMMON /BL29/   BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),              00000540
     1                BXT2(10),BE2(10)                                               00000550
      COMMON /BL30/   EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),           00000560
     1                ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)                     00000570
      COMMON /BL31/   DELSQ,EXT1(10)                                                 00000580
      COMMON /BL32/   TKN,ELAST,CHAR,SIGO                                            00000590
      COMMON /BL34/   DEE(4,4,200),DST(4,4,200)                                      00000600
      COMMON /BL100/  TEEO,$DYNMC                                                    00000610
      COMMON /BL101/  ZO(4,220),Z2(4,220),Z3(4,220),DELSD                            00000620
      COMMON /BL102/  DELOAD                                                         00000630
      COMMON /BL103/  MASS(200)                                                      00000640
      COMMON /BL104/  ZDOT(4,220)                                                    00000650
      COMMON /BL110/  TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),                 00000660
     1                QS(10)                                                         00000670
      COMMON /BL111/  ABZ,ABZO,ABZN,ABZ3,DD2                                         00000680
      COMMON /BLPLOT/  IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,             00000690
     1                 IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,             00000700
     2                 INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,                00000710
     3                 IPHIT,IPHI,$PLOTS,$MODAL                                     00000720
      COMMON /BLPLT1/  XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),              00000730
     1                 YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),             00000740
     2                 YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),             00000750
     3                 YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),             00000760
```

```
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),00000770
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),    00000780
     6                YPHI(100),XSTATN(100)                             00000790
      COMMON /BLDATA/ TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC            00000800
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                               00000810
C*****************************************************************00000820
      WRITE (6,8888)                                                    00000830
      DELSD=DELOAD*DELOAD                                               00000840
      KL=KMAX-1                                                         00000850
      KLL=KMAX-2                                                        00000860
      KMAX1=KMAX+1                                                      00000870
      KMAX2=KMAX+2                                                      00000880
      AK=KL                                                             00000890
      SIGT(1)=SIGO*TKN                                                  00000900
      SIGT(2)=SIGO/ELAST                                                00000910
      SIGC(1)=SIGO*CHAR/ELAST                                           00000920
      SIGC(2)=SIGO*TKN**3/CHAR                                          00000930
      IF (IBCINL.LT.0) GO TO 14                                         00000940
      DO 98 I=1,4                                                       00000950
      DO 98 J=1,4                                                       00000960
      KKLM=J/4+1                                                        00000970
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                  00000980
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                  00000990
   14 IF(IBCFNL.LT.0) GO TO 17                                          00001000
      DO 99 I=1,4                                                       00001010
      DO 99 J=1,4                                                       00001020
      KKLM=J/4+1                                                        00001030
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                  00001040
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                  00001050
   17 LAM=TKN/CHAR                                                      00001060
      SOE=SIGO/ELAST                                                    00001070
      OSE=.5*SOE                                                        00001080
      D1=1.0-NU                                                         00001090
      S1=1.0+NU                                                         00001100
      LAM2=LAM**2                                                       00001110
      IF(NDIMEN.LT.1) GO TO 228                                         00001120
      SIGO=1.0                                                          00001130
      ELAST=1.0                                                         00001140
      TKN=1.0                                                           00001150
      CHAR=1.0                                                          00001160
  228 DO 230 M=1,MAXM                                                   00001170
      N(M)=0.0                                                          00001180
      PX(M)=0.0                                                         00001190
      PT(M)=0.0                                                         00001200
      PR(M)=0.0                                                         00001210
      TT(M)=0.0                                                         00001220
      MT(M)=0.0                                                         00001230
      DT(M)=0.0                                                         00001240
```

```
      DMT(M)=0.0                                                          00001250
      MAXD(M)=0                                                           00001260
      MAXS(M)=0                                                           00001270
  230 MAXSY(M)=0                                                          00001280
      CALL GEOM                                                           00001290
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)          00001300
     1       +IABS(IRADII)                                                00001310
      ICHCK2=IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)          00001320
      IF (.NOT.$PLOTS) GO TO 1                                            00001330
      DO 2 K=1,KMAX                                                       00001340
    2 XSTATN(K)=FLOAT(K)                                                  00001350
      IF (ICHCK1.EQ.0) GO TO 1                                            00001360
      DO 1 K=1,KMAX                                                       00001370
      XRADII(K)=R(K)*CHAR                                                 00001380
      YGAMMA(K)=GAM(K)/CHAR                                               00001390
      YOMEGS(K)=OMXI(K)/CHAR                                              00001400
      YOMEGT(K)=OMT(K)/CHAR                                               00001410
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)                                      00001420
    1 CONTINUE                                                            00001430
      DO 86 K=1,KMAX                                                      00001440
   86 MASS(K)=0.                                                          00001450
      WRITE(6,802)                                                        00001460
      DO 978 K=1,KMAX                                                     00001470
      RKK=R(K)*CHAR                                                       00001480
      OMXIK=OMXI(K)/CHAR                                                  00001490
      GAMK=GAM(K)/CHAR                                                    00001500
      OMTK=OMT(K)/CHAR                                                    00001510
      DEOMXK=DEOMX(K)/(CHAR*CHAR)                                         00001520
  978 WRITE(6,803) K,RKK,GAMK,OMXIK,OMTK,DEOMXK                           00001530
  805 M0=0                                                                00001540
      M1=0                                                                00001550
      M2=0                                                                00001560
      M3=0                                                                00001570
      ABN=CHAR/SIGO/TKN                                                   00001580
      ZN=SIGO*TKN                                                         00001590
      WRITE(6,112)                                                        00001600
      DO 888 K=1,KMAX                                                     00001610
      CALL BDB(K,B,DB,D,DD)                                               00001620
      BST=ELAST*TKN                                                       00001630
      ZST=ELAST*TKN**3                                                    00001640
      B=B*BST                                                             00001650
      D=D*ZST                                                             00001660
      DB=DB/CHAR*BST                                                      00001670
      DD=DD/CHAR*ZST                                                      00001680
      WRITE (6,71) K,B,D,DB,DD                                            00001690
      IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                         00001700
      YBSTIF(K)=B                                                         00001710
      YDSTIF(K)=D                                                         00001720
```

```
      YBBSTF(K)=DB                                                          00001730
      YDDSTF(K)=DD                                                          00001740
  888 CONTINUE                                                              00001750
      CALL PLOAD(1)                                                         00001760
      CALL TLOAD(1)                                                         00001770
      DO 889 M=1,MNMAX                                                      00001780
      WRITE(6,113) N(M)                                                     00001790
      WRITE(6,114)                                                          00001800
      ICHCK3=IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)+IABS(IMT)              00001810
     1      +IABS(IDTT)+IABS(IDMT)                                          00001820
      DO 890 K=1,KMAX                                                       00001830
      CALL PLOAD(K)                                                         00001840
      CALL TLOAD(K)                                                         00001850
      PRM=PR(M)/ABN                                                         00001860
      PTM=PT(M)/ABN                                                         00001870
      PXM=PX(M)/ABN                                                         00001880
      TTM=TT(M)*ZN                                                          00001890
      EMTM=MT(M)/CHAR*ZN*TKN*TKN                                            00001900
      DTM=DT(M)/CHAR*ZN                                                     00001910
      DMTM=DMT(M)*ZN*TKN*TKN/(CHAR*CHAR)                                    00001920
      WRITE (6,115) K,PRM,PXM,PTM,TTM,EMTM,DTM,DMTM                         00001930
      IF (.NOT.$PLOTS.OR.(ICHCK3.EQ.0)) GO TO 890                           00001940
      YPR(K)=PRM                                                            00001950
      YPS(K)=PXM                                                            00001960
      YPT(K)=PTM                                                            00001970
      YTT(K)=TTM                                                            00001980
      YMT(K)=EMTM                                                           00001990
      YDTT(K)=DTM                                                           00002000
      YDMT(K)=DMTM                                                          00002010
  890 CONTINUE                                                              00002020
      IF (M.EQ.1) ICHCK3=ICHCK1+ICHCK2+ICHCK3                               00002030
      IF ($PLOTS.AND.(ICHCK3.GT.0)) CALL PLOT1(M)                           00002040
  889 CONTINUE                                                              00002050
      DELSQ=DEL**2                                                          00002060
      TDLI=.5/DEL                                                           00002070
      TDEL=2.0*DEL                                                          00002080
      MNINIT=1                                                              00002090
      MNMAXO=MNMAX                                                          00002100
      DO 23 I=1,4                                                           00002110
      DO 20 J=1,4                                                           00002120
      UNIT(I,J)=0.0                                                         00002130
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                              00002140
      NMAX=MAXM*KMAX2                                                       00002150
      DO 22 K=1,NMAX                                                        00002160
      DO 22 I=1,4                                                           00002170
      ZDOT(I,K)=0.0                                                         00002180
      ZO(I,K)=0.0                                                           00002190
      Z2(I,K)=0.0                                                           00002200
```

```
      Z3(I,K)=0.0                                                       00002210
   22 Z(I,K)=0.0                                                        00002220
      ALOAD=DELOAD                                                      00002230
      CALL PMATRX                                                       00002240
      LSTEP=1                                                           00002250
      LCHANG=0                                                          00002260
      ITR=1                                                             00002270
      ICORFL=0                                                          00002280
      IF(MNMAX.EQ.MAXM) ICORFL=1                                        00002290
      IPASS=0                                                           00002300
  400 CALL XANDZ                                                        00002310
      IF(ITRMAX.EQ.1) GO TO 50                                          00002320
      MNMAXO=MNMAX                                                      00002330
      IF(IPASS.LT.2) CALL MODES                                         00002340
      IF((NCONV.EQ.1).AND.(ITR.GT.1)) GO TO 50                          00002350
      IF(ITR.LT.ITRMAX) GO TO 23                                        00002360
      IF(LCHANG.LT.LCHMAX) GO TO 30                                     00002370
      WRITE(6,220) NO                                                   00002380
      GO TO 500                                                         00002390
   50 FL=LSTEP                                                          00002400
      FI=IPRINT                                                         00002410
      LI=LSTEP/IPRINT                                                   00002420
      FLI=LI                                                            00002430
      FT=FLI-FL/FI                                                      00002440
      IF(FT.EQ.0.) CALL OUTPUT(IMODE)                                   00002450
      IF(LSTEP.EQ.1) ITR=1                                              00002460
      IF(LSTEP.EQ.1) ITRPR=1                                            00002470
      IF(ITR.GT.ITRPR) ITRPR=ITR                                        00002480
      IF(LSTEP.GE.LSMAX) GO TO 360                                      00002490
   60 DO 61 MN=1,MNMAXO                                                 00002500
      DO 61 K=1,KMAX2                                                   00002510
      IK=K+(MN-1)*KMAX2                                                 00002520
      DO 61 I=1,4                                                       00002530
      ZN=2.0*Z(I,IK)-ZO(I,IK)                                           00002540
      ZO(I,IK)=Z(I,IK)                                                  00002550
   61 Z(I,IK)=ZN                                                        00002560
      IF(LSTEP.GE.LSMAX) GO TO 360                                      00002570
   62 ALOAD=ALOAD+DELOAD                                                00002580
      LSTEP=LSTEP+1                                                     00002590
      ITR=1                                                             00002600
      GO TO 400                                                         00002610
  360 WRITE(6,221) NO                                                   00002620
      GO TO 500                                                         00002630
   23 ITR=ITR+1                                                         00002640
      GO TO 400                                                         00002650
   30 IF(LSTEP-1) 310,310,320                                           00002660
  310 WRITE(6,223)                                                      00002670
      GO TO 500                                                         00002680
```

```
  320 WRITE(6,222)                                                      00002690
      LCHANG=LCHANG+1                                                   00002700
      LSTEP=LSTEP-1                                                     00002710
      ALOAD=ALOAD-DELOAD                                                00002720
      DELOAD=DELOAD/5.0                                                 00002730
      DO 32 MN=1,MNMAXO                                                 00002740
      DO 32 K=1,KMAX2                                                   00002750
      IK=K+(MN-1)*KMAX2                                                 00002760
      DO 32 I=1,4                                                       00002770
   32 Z(I,IK)=ZO(I,IK)                                                  00002780
      GO TO 62                                                          00002790
C*********************************************************************00002800
   71 FORMAT(20X,I3,4X,4E20.6)                                          00002810
  112 FORMAT(///17X,12H  STATION    20H       B STIFFNESS    20H       D ST00002820
     1IFFNESS    20H       B PRIME        20H       D PRIME        //)  00002830
  113 FORMAT(///25X,44HPRESSURE AND TEMPERATURE COEFFICIENTS FOR N=I3,8H00002840
     1  FOLLOW//)                                                       00002850
  114 FORMAT(5X,7HSTATION,3X,15H       PR        15H       PX        15H    00002860
     1  PT        15H       TT        15H       MT        15H       DTT    100002870
     25H       DMT       //)                                           00002880
  115 FORMAT(6X,I3,7X,7E15.4)                                           00002890
  220 FORMAT(1H1,80H            THE MAXIMUM NUMBER OF LOAD CHANGES HAVE BE00002900
     1EN MADE.  END PROBLEM NUMBERI4)                                   00002910
  221 FORMAT(1H1,79H            THE MAXIMUM NUMBER OF LOAD STEPS HAVE BEEN00002920
     1 TAKEN.  END PROBLEM NUMBERI4)                                    00002930
  222 FORMAT(1H1,119H           THE SOLUTION DID NOT CONVERGE WITHIN THE M00002940
     1AXIMUM NUMBER OF ITERATIONS.  THE LOAD FACTOR HAS BEEN DIVIDED BY 00002950
     25.)                                                               00002960
  223 FORMAT(1H1,69H            THE SOLUTION DID NOT CONVERGE FOR THE FIRS00002970
     1T LOAD INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR T00002980
     2RY A SMALLER VALUE FOR DELOAD.)                                   00002990
  802 FORMAT(1H1,17X,15H STATION        16H     RADIUS      16H     GAMMA 00003000
     1      16H     OMEGA S     16H     OMEGA THETA16H     DEOMEGA S  //) 00003010
  803 FORMAT(20X,I3, 9X,5E16.4)                                         00003020
 8888 FORMAT ('0',T20,'EXECUTING IN SUBROUTINE "STATIC"')               00003030
  500 RETURN                                                            00003040
      END                                                               00003050
```

```
      SUBROUTINE DYNAMC                                                 00000010
C*****************************************************************    00000020
C     THIS SUBROUTINE IS ONE OF THE MAJOR CONTROLLING SUBROUTINES FOR  *00000030
C     ALL DYNAMIC ANALYSIS PROBLEMS.  IT OPERATES IN A FASHION SIMILAR *00000040
C     TO SUBROUTINE STATIC.                                            *00000050
C*****************************************************************    00000060
      IMPLICIT LOGICAL*1 ($)                                            00000070
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                        00000080
      COMMON /IBL1/ MNMAX                                               00000090
      COMMON /IBL2/ N(10),MNINIT                                        00000100
C                   'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG      00000110
      COMMON /IBL3/ M0,M1,M2,M3                                         00000120
      COMMON /IBL4/ KMAX,KL                                             00000130
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000140
      COMMON /IBL6/ KLL                                                 00000150
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       00000160
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               00000170
      COMMON /IBL8/ LSTEP,ITR                                           00000180
      COMMON /IBL9/ MAXM                                                00000190
      COMMON /IBL10/ IFREQ,NTHMAX                                       00000200
      COMMON /IBL11/ ICORFL,IPASS                                       00000210
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000220
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000230
      COMMON /BL1/   A(4,4),BEE(4,4),C(4,4)                             00000240
      COMMON /BL3/   PR(10),PX(10),PT(10)                               00000250
      COMMON /BL4/   P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),     00000260
     1               ZF3M(4,4,10),ZF4M(4,4,10)                          00000270
      COMMON /BL5/   TT(10),MT(10),DT(10),DMT(10)                       00000280
C                   'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPOL & FNLPOL00000290
      COMMON /BL6/   Z(4,220),SOE,OSE,ALOAD                             00000300
      COMMON /BL7/   D1,S1                                              00000310
      COMMON /BL8/   R(200),GAM(200),OMT(200)                           00000320
      COMMON /BL9/   FFS(4,10),EL1S(4),GEES(4,10)                       00000330
      COMMON /BL10/  PHIX(10),PHIT(10),PHI(10)                          00000340
      COMMON /BL11/  OMXI(200),PHEE,T0,T2                               00000350
      COMMON /BL12/  TDLI,TDEL                                          00000360
      COMMON /BL13/  OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),       00000370
     1               UNIT(4,4)                                          00000380
      COMMON /BL14/  LAM2,LSD18,LSD1N                                   00000390
      COMMON /BL15/  NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),00000400
     1               V3(10),W3(10)                                      00000410
      COMMON /BL16/  EPS                                                00000420
      COMMON /BL17/  DEL                                                00000430
      COMMON /BL18/  EL1(4),ELL(4)                                      00000440
      COMMON /BL19/  TH(36)                                             00000450
      COMMON /BL20/  DEOMX(200)                                         00000460
      COMMON /BL23/  JAY(4,4),H(4,4)                                    00000470
      COMMON /BL24/  DL(4,4,10),DG(4,4,10),DF(4,4,10)                   00000480
```

```
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                    00000490
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                        00000500
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)     00000510
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),       00000520
     1              BXT2(10),BE2(10)                                        00000530
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),    00000540
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)              00000550
      COMMON /BL31/ DELSQ,EXT1(10)                                          00000560
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                     00000570
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                               00000580
      COMMON /BL100/ TEEO,$DYNMC                                            00000590
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                    00000600
      COMMON /BL102/ DELOAD                                                 00000610
      COMMON /BL103/ MASS(200)                                              00000620
      COMMON /BL104/ ZDOT(4,220)                                            00000630
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),         00000640
     1               QS(10)                                                 00000650
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                                 00000660
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,     00000670
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,      00000680
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,         00000690
     3                IPHIT,IPHI,$PLOTS,$MODAL                              00000700
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),      00000710
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),      00000720
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),      00000730
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),      00000740
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),    00000750
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),        00000760
     6                YPHI(100),XSTATN(100)                                 00000770
      COMMON /BLDATA/ TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC                00000780
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                                   00000790
      DIMENSION VBAR(10),AVB(10)                                            00000800
C*********************************************************************    00000810
      WRITE (6,8888)                                                        00000820
      DELSD=DELOAD*DELOAD                                                   00000830
      KL=KMAX-1                                                             00000840
      KLL=KMAX-2                                                            00000850
      KMAX1=KMAX+1                                                          00000860
      KMAX2=KMAX+2                                                          00000870
      AK=KL                                                                 00000880
      SIGT(1)=SIGO*TKN                                                      00000890
      SIGT(2)=SIGO/ELAST                                                    00000900
      SIGC(1)=SIGO*CHAR/ELAST                                               00000910
      SIGC(2)=SIGO*TKN**3/CHAR                                              00000920
      IF (IBCINL.LT.0) GO TO 14                                             00000930
      DO 98 I=1,4                                                           00000940
      DO 98 J=1,4                                                          000000950
      KKLM=J/4+1                                                            00000960
```

```
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                  00000970
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                  00000980
   14 IF(IBCFNL.LT.0) GO TO 17                                          00000990
      DO 99 I=1,4                                                       00001000
      DO 99 J=1,4                                                       00001010
      KKLM=J/4+1                                                        00001020
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                  00001030
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                  00001040
   17 LAM=TKN/CHAR                                                      00001050
      SOE=SIGO/ELAST                                                    00001060
      OSE=.5*SOE                                                        00001070
      D1=1.0-NU                                                         00001080
      S1=1.0+NU                                                         00001090
      LAM2=LAM**2                                                       00001100
      IF(NDIMEN.LT.1) GO TO 228                                         00001110
      SIGO=1.0                                                          00001120
      ELAST=1.0                                                         00001130
      TKN=1.0                                                           00001140
      CHAR=1.0                                                          00001150
  228 DO 230 M=1,MAXM                                                   00001160
      N(M)=0.0                                                          00001170
      PX(M)=0.0                                                         00001180
      PT(M)=0.0                                                         00001190
      PR(M)=0.0                                                         00001200
      TT(M)=0.0                                                         00001210
      MT(M)=0.0                                                         00001220
      DT(M)=0.0                                                         00001230
      DMT(M)=0.0                                                        00001240
      MAXD(M)=0                                                         00001250
      MAXS(M)=0                                                         00001260
  230 MAXSY(M)=0                                                        00001270
      CALL GEOM                                                         00001280
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)        00001290
     1       +IABS(IRADII)                                              00001300
      ICHCK2=IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)        00001310
      IF (.NOT.$PLOTS) GO TO 1                                          00001320
      DO 2 K=1,KMAX                                                     00001330
    2 XSTATN(K)=FLOAT(K)                                                00001340
      IF (ICHCK1.EQ.0) GO TO 1                                          00001350
      DO 1 K=1,KMAX                                                     00001360
      XRADII(K)=R(K)*CHAR                                               00001370
      YGAMMA(K)=GAM(K)/CHAR                                             00001380
      YOMEGS(K)=OMXI(K)/CHAR                                            00001390
      YOMEGT(K)=OMT(K)/CHAR                                             00001400
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)                                    00001410
    1 CONTINUE                                                          00001420
  804 WRITE(6,810)                                                      00001430
      TEED=TEEO                                                         00001440
```

```
      IF(NDIMEN.EQ.1) TEED=1.0                                          00001450
      DO 979 K=1,KMAX                                                   00001460
      RKK=R(K)*CHAR                                                     00001470
      OMXIK=OMXI(K)/CHAR                                                00001480
      GAMK=GAM(K)/CHAR                                                  00001490
      OMTK=OMT(K)/CHAR                                                  00001500
      DEOMXK=DEOMX(K)/(CHAR*CHAR)                                       00001510
      AMSS=MASS(K)*TEED**2*ELAST*TKN/CHAR**2                            00001520
  979 WRITE(6,813) K,RKK,GAMK,OMXIK,OMTK,DEOMXK,AMSS                    00001530
  805 M0=0                                                              00001540
      M1=0                                                              00001550
      M2=0                                                              00001560
      M3=0                                                              00001570
      ABN=CHAR/SIGO/TKN                                                 00001580
      ZN=SIGO*TKN                                                       00001590
      WRITE(6,112)                                                      00001600
      DO 888 K=1,KMAX                                                   00001610
      CALL BDB(K,B,DB,D,DD)                                             00001620
      BST=ELAST*TKN                                                     00001630
      ZST=ELAST*TKN**3                                                  00001640
      B=B*BST                                                           00001650
      D=D*ZST                                                           00001660
      DB=DB/CHAR*BST                                                    00001670
      DD=DD/CHAR*ZST                                                    00001680
      WRITE (6,71) K,B,D,DB,DD                                          00001690
      IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                       00001700
      YBSTIF(K)=B                                                       00001710
      YDSTIF(K)=D                                                       00001720
      YBBSTF(K)=DB                                                      00001730
      YDDSTF(K)=DD                                                      00001740
  888 CONTINUE                                                          00001750
      CALL PLOAD(1)                                                     00001760
      CALL TLOAD(1)                                                     00001770
      DELSQ=DEL**2                                                      00001780
      TDLI=.5/DEL                                                       00001790
      TDEL=2.0*DEL                                                      00001800
      MNINIT=1                                                          00001810
      MNMAXO=MNMAX                                                      00001820
      DO 20 I=1,4                                                       00001830
      DO 20 J=1,4                                                       00001840
      UNIT(I,J)=0.0                                                     00001850
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                          00001860
      NMAX=MAXM*KMAX2                                                   00001870
      DO 22 K=1,NMAX                                                    00001880
      DO 22 I=1,4                                                       00001890
      ZDOT(I,K)=0.0                                                     00001900
      ZO(I,K)=0.0                                                       00001910
      Z2(I,K)=0.0                                                       00001920
```

```
      Z3(I,K)=0.0                                                       00001930
   22 Z(I,K)=0.0                                                        00001940
      IF(IC.EQ.0) GO TO 834                                             00001950
      CALL INITL                                                        00001960
      ACO=CHAR*SIGO/ELAST                                               00001970
      ACM=SIGO*TKN**3/CHAR                                              00001980
      DO 830 M=1,MNMAX                                                  00001990
      MM=(M-1)*KMAX2                                                    00002000
      WRITE(6,126) N(M)                                                 00002010
      WRITE(6,127)                                                      00002020
      DO 831 K=2,KMAX1                                                  00002030
      MK=K+MM                                                           00002040
      TU=ACO*ZO(1,MK)                                                   00002050
      TV=ACO*ZO(2,MK)                                                   00002060
      TW=ACO*ZO(3,MK)                                                   00002070
      TM=ACM*ZO(4,MK)                                                   00002080
      KK=K-1                                                            00002090
  831 WRITE(6,71) KK,TU,TV,TW,TM                                        00002100
      WRITE(6,129)                                                      00002110
      DO 830 K=2,KMAX1                                                  00002120
      ACD=CHAR*SIGO/(ELAST*TEEO)                                        00002130
      AMD=SIGO*TKN**3/(CHAR*TEEO)                                       00002140
      MK=K+MM                                                           00002150
      TU=ACD*ZDOT(1,MK)                                                 00002160
      TV=ACD*ZDOT(2,MK)                                                 00002170
      TW=ACD*ZDOT(3,MK)                                                 00002180
      TM=AMD*ZDOT(4,MK)                                                 00002190
      KK=K-1                                                            00002200
      WRITE(6,71) KK,TU,TV,TW,TM                                        00002210
      DO 830 I=1,4                                                      00002220
      Z(I,MK)=ZO(I,MK)+ZDOT(I,MK)*DELOAD                                00002230
      Z2(I,MK)=ZO(I,MK)-ZDOT(I,MK)*DELOAD                               00002240
  830 Z3(I,MK)=ZO(I,MK)-2.*ZDOT(I,MK)*DELOAD                            00002250
  834 CONTINUE                                                          00002260
      ALOAD=1.0                                                         00002270
      CALL PMATRX                                                       00002280
      LSTEP=1                                                           00002290
      LCHANG=0                                                          00002300
      ITR=1                                                             00002310
      ICORFL=0                                                          00002320
      IF(MNMAX.EQ.MAXM) ICORFL=1                                        00002330
      IPASS=0                                                           00002340
  400 CALL XANDZ                                                        00002350
      IF(ITRMAX.EQ.1) GO TO 50                                          00002360
      MNMAXO=MNMAX                                                      00002370
      IF(IPASS.LT.2) CALL MODES                                         00002380
      IF(NCONV.EQ.1) GO TO 50                                           00002390
      IF(ITR.LT.ITRMAX) GO TO 23                                        00002400
```

```
      GO TO 365                                                                 00002410
   50 FL=LSTEP                                                                  00002420
      FI=IPRINT                                                                 00002430
      LI=LSTEP/IPRINT                                                           00002440
      FLI=LI                                                                    00002450
      FT=FLI-FL/FI                                                              00002460
      IF(FT.EQ.0.) CALL OUTPUT(IMODE)                                           00002470
      IF(LSTEP.EQ.1) ITR=1                                                      00002480
      IF(LSTEP.EQ.1) ITRPR=1                                                    00002490
      IF(ITR.GT.ITRPR) ITRPR=ITR                                                00002500
      IF(LSTEP.GE.LSMAX) GO TO 360                                              00002510
      DO 65 MN=1,MNMAXO                                                         00002520
      DO 65 K=1,KMAX2                                                           00002530
      IK=K+(MN-1)*KMAX2                                                         00002540
      DO 65 I=1,4                                                               00002550
      ZN=3.0*(Z(I,IK)-ZO(I,IK))+Z2(I,IK)                                        00002560
      Z3(I,IK)=Z2(I,IK)                                                         00002570
      Z2(I,IK)=ZO(I,IK)                                                         00002580
      ZO(I,IK)=Z(I,IK)                                                          00002590
   65 Z(I,IK)=ZN                                                                00002600
      ALOAD=1.0                                                                 00002610
      LSTEP=LSTEP+1                                                             00002620
      ITR=1                                                                     00002630
      GO TO 400                                                                 00002640
   23 ITR=ITR+1                                                                 00002650
      GO TO 400                                                                 00002660
  360 WRITE(6,271)                                                              00002670
      WRITE(6,188)(AVB(M),M=1,MAXM)                                             00002680
      WRITE(6,189) ITRPR                                                        00002690
      GO TO 500                                                                 00002700
  365 IF(LSTEP.EQ.1) GO TO 367                                                  00002710
      WRITE(6,266) ITRMAX,LSTEP,NO                                              00002720
      WRITE(6,188)(AVB(M),M=1,MAXM)                                             00002730
      GO TO 500                                                                 00002740
  367 WRITE(6,273)                                                              00002750
      GO TO 500                                                                 00002760
C*****************************************************************************00002770
   71 FORMAT(20X,I3,4X,4E20.6)                                                  00002780
  112 FORMAT(////17X,12H  STATION    20H      B STIFFNESS    20H      D ST00002790
     1IFFNESS    20H      B PRIME       20H      D PRIME       //)            00002800
  126 FORMAT(////5X,29HTHE INITIAL CONDITIONS FOR N=I3,8H  FOLLOW//)           00002810
  127 FORMAT(19X,7HSTATION,3X,20H      U              20H      V              00002820
     1    20H      W              20H      M S            //)                00002830
  129 FORMAT(///19X,7HSTATION,3X,20H      U DOT          20H      V DOT       00002840
     1    20H      W DOT          20H      M S DOT        //)                00002850
  188 FORMAT(////' THE MAXIMUM VBAR FOR EACH MODE IS'/10E11.4)                  00002860
  189 FORMAT(/'  THE MAXIMUMUM NUMBER OF ITERATIONS TAKEN IS  'I3)             00002870
  266 FORMAT(1H1,35H   THE SOLUTION DID NOT CONVERGE INI3,24H ITERATIONS00002880
```

```
     1 AT TIME STEPI5,21H.  END PROBLEM NUMBERI4,1H.)                    00002890
  271 FORMAT(1H1,79H          THE MAXIMUM NUMBER OF TIME STEPS HAVE BEEN00002900
     1 TAKEN.  END PROBLEM NUMBERI4)                                     00002910
  273 FORMAT(1H1,69H          THE SOLUTION DID NOT CONVERGE FOR THE FIRS00002920
     1T TIME INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR T00002930
     2RY A SMALLER VALUE FOR DELOAD.)                                    00002940
  810 FORMAT(1H1, 5X,15H STATION       16H     RADIUS     16H     GAMMA 00002950
     1     16H     OMEGA S    16H     OMEGA THETA16H     DEOMEGA S  16H 00002960
     2    MASS        //)                                               00002970
  813 FORMAT( 8X,I3, 9X,6E16.4)                                          00002980
 8888 FORMAT (' ',T10,'EXECUTING IN SUBROUTINE "DYNAMIC"')               00002990
  500 RETURN                                                            00003000
      END                                                               00003010
```



```
      SUBROUTINE OUTPUT(IMODE)                                          00000010
C*****************************************************************     00000020
C         THIS SUBROUTINE PREPARES THE PRINTOUT MATERIAL.  EVERY IPRINT *00000030
C         CONVERGED SOLUTION IS PRINTED.  THE FOURIER COEFFICIENTS OF THE00000040
C         INPLANE FORCES, MERIDIONAL TRANSVERSE FORCE, CIRCUMFERENTIAL  *00000050
C         BENDING MOMENT, TWISTING MOMENT AND ROTATIONS CAN BE COMPUTED *00000060
C         AND PRINTED WITH THE SOLUTION Z FOR THE FOURIER COEFFICIENTS  *00000070
C         OF THE THREE DISPLACEMENTS AND MERIDIONAL BENDING MOMENT.  THIS00000080
C         OUTPUT MATERIAL IS CONVERTED FROM DIMENSIONLESS FORM TO DIMEN-*00000090
C         SIONAL FORM HERE.  PROVISION IS MADE TO PRINT ONLY AT STATIONS*00000100
C         1, IFREQ+1, 2IFREQ+1, ETC.  THIS SUBROUTINE ALSO PERFORMS THE *00000110
C         SUMMATION PROCESS FOR COMPUTING THE TOTAL VALUES OF THE FORCES,00000120
C         MOMENTS, DISPLACEMENTS AND ROTATIONS AT THE NTHMAX POSITIONS  *00000130
C         AROUND THE CIRCUMFERENCE PRESCRIBED IN THE INPUT DATA.        *00000140
C      DATA TO BE PLOTTED IS PREPARED HERE, IF REQ'D                    *00000150
C*****************************************************************     00000160
      IMPLICIT LOGICAL*1 ($)                                            00000170
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                 00000180
      COMMON /IBL2/ N(10),MNINIT                                        00000190
      COMMON /IBL3/ M0,M1,M2,M3                                         00000200
      COMMON /IBL4/ KMAX,KL                                             00000210
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000220
      COMMON /IBL7/ MNMAX0,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       00000230
```

```
     1                JS(10,10),ID(10,10),JD(10,10),IJS(10)                  00000240
      COMMON /IBL8/ LSTEP,ITR                                                00000250
      COMMON /IBL10/ IFREQ,NTHMAX                                            00000260
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                       00000270
      COMMON /IBL13/ ITRMAX,LSMAX                                            00000280
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),           00000290
     1              ZF3M(4,4,10),ZF4M(4,4,10)                               00000300
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                             00000310
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                                   00000320
      COMMON /BL7/  D1,S1                                                    00000330
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                 00000340
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                                00000350
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                     00000360
      COMMON /BL12/ TDLI,TDEL                                                00000370
      COMMON /BL14/ LAM2,LSD18,LSD1N                                         00000380
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),     00000390
     1              V3(10),W3(10)                                            00000400
      COMMON /BL17/ DEL                                                      00000410
      COMMON /BL19/ TH(36)                                                   00000420
      COMMON /BL20/ DEOMX(200)                                               00000430
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                         00000440
      COMMON /BL31/ DELSQ,EXT1(10)                                           00000450
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                      00000460
      COMMON /BL100/ TEEO,$DYNMC                                             00000470
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                     00000480
      COMMON /BL102/ DELOAD                                                  00000490
      COMMON /BL103/ MASS(200)                                               00000500
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),          00000510
     1               QS(10)                                                  00000520
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                                  00000530
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,      00000540
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,       00000550
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,          00000560
     3                IPHIT,IPHI,$PLOTS,$MODAL                               00000570
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),       00000580
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),       00000590
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),       00000600
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),       00000610
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),     00000620
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),         00000630
     6                YPHI(100),XSTATN(100)                                  00000640
      DIMENSION PTF(200),PF(200)                                             00000650
C*****************************************************************          00000660
      ABZO=SIGO/ELAST                                                        00000670
      IF ($DYNMC) GO TO 181                                                  00000680
      WRITE(6,101) LSTEP,ALOAD,ITR                                           00000690
      GO TO 182                                                              00000700
  181 TI=LSTEP*DELOAD                                                        00000710
```

```
      DTI=TI*TEEO                                                        00000720
      WRITE(6,151) LSTEP,TI,DTI,ITR                                      00000730
  182 LAM=TKN/CHAR                                                       00000740
      ENL=1.                                                             00000750
      ABZ=SIGO*TKN                                                       00000760
      ABZ3=ABZ*TKN*TKN/CHAR                                              00000770
      ABZN=CHAR*SIGO/ELAST                                               00000780
      IF(ITRMAX.EQ.1) ENL=0.                                             00000790
      DD2=1.-NU**2                                                       00000800
      D2I=1./DD2                                                         00000810
      DPI=1./S1                                                          00000820
      DNI=1./D1                                                          00000830
      TDLSQI=.5/DELSQ                                                    00000840
      ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)        00000850
     1       +IABS(IMTH)+IABS(IMSTH)                                     00000860
      ICHCK2=IABS(IU)+IABS(IV)+IABS(IW)+IABS(IPHIS)+IABS(IPHIT)          00000870
     1       +IABS(IPHI)                                                 00000880
      IF(NTHMAX.EQ.0) GO TO 991                                          00000890
      DO 21 NTH=1,NTHMAX                                                 00000900
      DO 1 MN=1,MNMAXO                                                   00000910
      I1=1+(MN-1)*KMAX2                                                  00000920
      I2=I1+1                                                            00000930
      U1(MN)=Z(1,I1)                                                     00000940
      U2(MN)=Z(1,I2)                                                     00000950
      V1(MN)=Z(2,I1)                                                     00000960
      V2(MN)=Z(2,I2)                                                     00000970
      W1(MN)=Z(3,I1)                                                     00000980
    1 W2(MN)=Z(3,I2)                                                     00000990
      THET=TH(NTH)                                                       00001000
      WRITE(6,116) THET                                                  00001010
      DO 121 K=1,KMAX                                                    00001020
      K1=K+1                                                             00001030
      CALL BDB(K,BS,DB,DS,DD)                                            00001040
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                            00001050
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 999                               00001060
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)                         00001070
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 999                            00001080
      CALL PHIBET(K)                                                     00001090
      DEX=DEOMX(K)                                                       00001100
      RRA=1./R(K)                                                        00001110
      OX=OMXI(K)                                                         00001120
      OT=OMT(K)                                                          00001130
      GA=GAM(K)                                                          00001140
      DOXT=OX-OT                                                         00001150
      GDO=GA*DOXT                                                        00001160
      DD2D=DD2*DS                                                        00001170
      DO 3 MN=1,MNMAXO                                                   00001180
      EN=N(MN)                                                           00001190
```

```
      ENR=EN*RRA                                                                00001200
      CALL TLOAD(K)                                                             00001210
      TTS=TT(MN)*ALOAD                                                          00001220
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))               00001230
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))               00001240
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SOE*BXT3(MN)        00001250
     1)                                                                         00001260
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                               00001270
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)         00001280
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DOXT*PHI(MN))                   00001290
      TX(MN)=BS*(EX+NU*ET)-TTS                                                  00001300
      TTH(MN)=BS*(ET+NU*EX)-TTS                                                 00001310
      TXT(MN)=BS*D1*EXT                                                         00001320
      MK1=K1+(MN-1)*KMAX2                                                       00001330
      MX(MN)=Z(4,MK1)                                                           00001340
      MTH(MN)=NU*MX(MN)+DD2D*KT-D1*MT(MN)*ALOAD                                 00001350
      MXT(MN)=DS*D1*KXT                                                         00001360
      MK11=MK1+1                                                                00001370
      MKK1=MK1-1                                                                00001380
      QS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MK11)-Z(4,MKK1))*TDLI                00001390
     1      +ENR*MXT(MN)-GA*MTH(MN))                                            00001400
      MX(MN)=MX(MN)*ABZ3                                                        00001410
      MTH(MN)=MTH(MN)*ABZ3                                                      00001420
      MXT(MN)=MXT(MN)*ABZ3                                                      00001430
      TX(MN)=TX(MN)*ABZ                                                         00001440
      TTH(MN)=TTH(MN)*ABZ                                                       00001450
      TXT(MN)=TXT(MN)*ABZ                                                       00001460
      PHIX(MN)=PHIX(MN)*ABZO                                                    00001470
      PHIT(MN)=PHIT(MN)*ABZO                                                    00001480
      PHI(MN)=PHI(MN)*ABZO                                                      00001490
      U1(MN)=U2(MN)                                                             00001500
      U2(MN)=U3(MN)                                                             00001510
      V1(MN)=V2(MN)                                                             00001520
      V2(MN)=V3(MN)                                                             00001530
      W1(MN)=W2(MN)                                                             00001540
    3 W2(MN)=W3(MN)                                                             00001550
      FK=K-1                                                                    00001560
      FIFREQ=IFREQ                                                              00001570
      KTST=(K-1)/IFREQ                                                          00001580
      FKTST=KTST                                                                00001590
      FKTEST=FK/FIFREQ-FKTST                                                    00001600
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999                                         00001610
      IF(FKTEST.NE.0.) GO TO 2                                                  00001620
  999 X(1,K)=0.                                                                 00001630
      X(2,K)=0.                                                                 00001640
      X(3,K)=0.                                                                 00001650
      X(4,K)=0.                                                                 00001660
      PTF(K)=0.                                                                 00001670
```

```
      PF(K)=0.                                                                  00001680
      AMX=0.                                                                    00001690
      AMTH=0.                                                                   00001700
      AMXTH=0.                                                                  00001710
      ANX=0.                                                                    00001720
      ANTH=0.                                                                   00001730
      ANXTH=0.                                                                  00001740
      AQS=0.                                                                    00001750
      DO 72 MN=1,MNMAXO                                                         00001760
      EN=N(MN)                                                                  00001770
      FC=EN*THET                                                                00001780
      SN=SIN(FC)                                                                00001790
      CS=COS(FC)                                                                00001800
      X(1,K)=X(1,K)+U1(MN)*CS*ABZN                                              00001810
      X(2,K)=X(2,K)+V1(MN)*SN*ABZN                                              00001820
      X(3,K)=X(3,K)+W1(MN)*CS*ABZN                                              00001830
      X(4,K)=X(4,K)+PHIX(MN)*CS                                                 00001840
      PTF(K)=PTF(K)+PHIT(MN)*SN                                                 00001850
      AMX=AMX+MX(MN)*CS                                                         00001860
      AMTH=AMTH+MTH(MN)*CS                                                      00001870
      AMXTH=AMXTH+MXT(MN)*SN                                                    00001880
      ANX=ANX+TX(MN)*CS                                                         00001890
      ANTH=ANTH+TTH(MN)*CS                                                      00001900
      AQS=AQS+QS(MN)*CS                                                         00001910
      ANXTH=ANXTH+TXT(MN)*SN                                                    00001920
   72 PF(K)=PF(K)+PHI(MN)*SN                                                    00001930
  429 CONTINUE                                                                  00001940
      IF(K.EQ.1) WRITE(6,117)                                                   00001950
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))               00001951
     1 GO TO 220                                                                00001952
      GO TO 221                                                                 00001953
  220 WRITE (6,1181) K,ANX,ANTH,ANXTH,AMX,AMTH,AMXTH                            00001954
      GO TO 1354                                                                00001955
  221 WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH                          00001960
 1354 IF ($MODAL.OR.(ICHCK1.EQ.0)) GO TO 2                                      00001970
      YNS(K)=ANX                                                                00001980
      YNTH(K)=ANTH                                                              00001990
      YNSTH(K)=ANXTH                                                            00002000
      YQS(K)=AQS                                                                00002010
      YMS(K)=AMX                                                                00002020
      YMTH(K)=AMTH                                                              00002030
      YMSTH(K)=AMXTH                                                            00002040
    2 CONTINUE                                                                  00002050
  121 CONTINUE                                                                  00002060
      DO 663 K=1,KMAX                                                           00002070
      FK=K-1                                                                    00002080
      FIFREQ=IFREQ                                                              00002090
      KTST=(K-1)/IFREQ                                                          00002100
```

```
      FKTST=KTST                                                        00002110
      FKTEST=FK/FIFREQ-FKTST                                            00002120
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 661                                 00002130
      IF(FKTEST.NE.0.) GO TO 658                                        00002140
  661 IF(K.EQ.1) WRITE(6,217)                                           00002150
      WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),PTF(K),PF(K)           00002160
      IF ($MODAL.OR.(ICHCK2.EQ.0)) GO TO 658                            00002170
      YU(K)=X(1,K)                                                      00002180
      YV(K)=X(2,K)                                                      00002190
      YW(K)=X(3,K)                                                      00002200
      YPHIS(K)=X(4,K)                                                   00002210
      YPHIT(K)=PTF(K)                                                   00002220
      YPHI(K)=PF(K)                                                     00002230
  658 DO 659 I=1,4                                                      00002240
  659 X(I,K)=0.                                                         00002250
  660 CONTINUE                                                          00002260
      IF ($PLOTS.AND..NOT.$MODAL.AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))  00002270
     1   CALL PLOT2(NTH)                                                00002280
   21 CONTINUE                                                          00002290
  991 IF(IMODE.LE.0) RETURN                                             00002300
      DO 534 MN=1,MNMAXO                                                00002310
      WRITE(6,749) N(MN)                                                00002320
      DO 521 MM=1,MNMAXO                                                00002330
      I1=1+(MM-1)*KMAX2                                                 00002340
      I2=I1+1                                                           00002350
      U1(MM)=Z(1,I1)                                                    00002360
      U2(MM)=Z(1,I2)                                                    00002370
      V1(MM)=Z(2,I1)                                                    00002380
      V2(MM)=Z(2,I2)                                                    00002390
      W1(MM)=Z(3,I1)                                                    00002400
      W2(MM)=Z(3,I2)                                                    00002410
  521 CONTINUE                                                          00002420
      DO 445 K=1, KMAX                                                  00002430
      K1=K+1                                                            00002440
      CALL BDB(K,BS,DB,DS,DD)                                           00002450
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                           00002460
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)                        00002470
      TXZ=TX(MN)                                                        00002480
      TTHZ=TTH(MN)                                                      00002490
      TXTZ=TXT(MN)                                                      00002500
      AMXZ=MX(MN)                                                       00002510
      AMTHZ=MTH(MN)                                                     00002520
      AMXTZ=MXT(MN)                                                     00002530
      QSZ=QS(MN)                                                        00002540
      X(1,K)=PHIX(MN)                                                   00002550
      X(2,K)=PHIT(MN)                                                   00002560
      X(3,K)=PHI(MN)                                                    00002570
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 583                              00002580
```

```
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 583                                   00002590
      CALL PHIBET(K)                                                            00002600
      DEX=DEOMX(K)                                                              00002610
      RRA=1./R(K)                                                               00002620
      OX=OMXI(K)                                                                00002630
      OT=OMT(K)                                                                 00002640
      GA=GAM(K)                                                                 00002650
      DOXT=OX-OT                                                                00002660
      GDO=GA*DOXT                                                               00002670
      DD2D=DD2*DS                                                               00002680
      EN=N(MN)                                                                  00002690
      ENR=EN*RRA                                                                00002700
      CALL TLOAD(K)                                                             00002710
      TTS=TT(MN)*ALOAD                                                          00002720
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))               00002730
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))               00002740
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SOE*BXT3(MN)00002750
     1)                                                                         00002760
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                               00002770
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)         00002780
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DOXT*PHI(MN))                   00002790
      TXZ=(BS*(EX+NU*ET)-TTS)*ABZ                                               00002800
      TTHZ=(BS*(ET+NU*EX)-TTS)*ABZ                                              00002810
      TXTZ=BS*D1*EXT*ABZ                                                        00002820
      MK1=K1+(MN-1)*KMAX2                                                       00002830
      AMXZ=Z(4,MK1)                                                             00002840
      AMTHZ=NU*AMXZ+DD2D*KT-D1*MT(MN)*ALOAD                                     00002850
      AMXTZ=DS*D1*KXT                                                           00002860
      MK11=MK1+1                                                                00002870
      MKK1=MK1-1                                                                00002880
      QSZ=SIGO*TKN*LAM2*(GA*AMXZ+(Z(4,MK11)-Z(4,MKK1))*TDLI+ENR*AMXTZ           00002890
     1     -GA*AMTHZ)                                                           00002900
      AMXZ=AMXZ*ABZ3                                                            00002910
      AMTHZ=AMTHZ*ABZ3                                                          00002920
      AMXTZ=AMXTZ*ABZ3                                                          00002930
      X(1,K)=PHIX(MN)*ABZO                                                      00002940
      X(2,K)=PHIT(MN)*ABZO                                                      00002950
      X(3,K)=PHI(MN)*ABZO                                                       00002960
      DO 533 MM=1,MNMAXO                                                        00002970
      U1(MM)=U2(MM)                                                             00002980
      U2(MM)=U3(MM)                                                             00002990
      V1(MM)=V2(MM)                                                             00003000
      V2(MM)=V3(MM)                                                             00003010
      W1(MM)=W2(MM)                                                             00003020
  533 W2(MM)=W3(MM)                                                             00003030
      FK=K-1                                                                    00003040
      FIFREQ=IFREQ                                                              00003050
      KTST=(K-1)/IFREQ                                                          00003060
```

```
      FKTST=KTST                                                          00003070
      FKTEST=FK/FIFREQ-FKTST                                              00003080
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583                                   00003090
      IF(FKTEST.NE.0.) GO TO 445                                          00003100
  583 CONTINUE                                                            00003110
      IF(K.EQ.1) WRITE(6,117)                                             00003120
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))         00003121
     1 GO TO 4444                                                         00003122
      GO TO 4445                                                          00003123
 4444 WRITE (6,1181) K,TXZ,TTHZ,TXTZ,AMXZ,AMTHZ,AMXTZ                    00003124
      GO TO 4446                                                          00003125
 4445 WRITE(6,118) K,TXZ,TTHZ,TXTZ,QSZ,AMXZ,AMTHZ,AMXTZ                  00003130
 4446 IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK1.EQ.0)) GO TO 445          00003140
      YNS(K)=TXZ                                                          00003150
      YNTH(K)=TTHZ                                                        00003160
      YNSTH(K)=TXTZ                                                       00003170
      YQS(K)=QSZ                                                          00003180
      YMS(K)=AMXZ                                                         00003190
      YMTH(K)=AMTHZ                                                       00003200
      YMSTH(K)=AMXTZ                                                      00003210
  445 CONTINUE                                                            00003220
  446 WRITE(6,217)                                                        00003230
      DO 447 K=1,KMAX                                                     00003240
      FK=K-1                                                              00003250
      FIFREQ=IFREQ                                                        00003260
      KTST=(K-1)/IFREQ                                                    00003270
      FKTST=KTST                                                          00003280
      FKTEST=FK/FIFREQ-FKTST                                              00003290
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 593                                   00003300
      IF(FKTEST.NE.0.) GO TO 446                                          00003310
  593 KZ=K+1+(MN-1)*KMAX2                                                 00003320
      UP=Z(1,KZ)*ABZN                                                     00003330
      VP=Z(2,KZ)*ABZN                                                     00003340
      WP=Z(3,KZ)*ABZN                                                     00003350
      WRITE(6,218) K,UP,VP,WP,X(1,K),X(2,K),X(3,K)                        00003360
      IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK2.EQ.0)) GO TO 447          00003370
      YU(K)=UP                                                            00003380
      YV(K)=VP                                                            00003390
      YW(K)=WP                                                            00003400
      YPHIS(K)=X(1,K)                                                     00003410
      YPHIT(K)=X(2,K)                                                     00003420
      YPHI(K)=X(3,K)                                                      00003430
  447 CONTINUE                                                            00003440
      IF ($PLOTS.AND.$MODAL.AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))         00003450
     1   CALL PLOT2(1)                                                    00003460
  534 CONTINUE                                                            00003470
C*************************************************************************00003480
  101 FORMAT('1','         THE LOAD STEP NUMBER IS ',I2,'          THE L00003490
```

```
     1OAD FACTOR IS ',E11.4,'          THE SOLUTION CONVERGED IN ',I2,' 00003500
     2ITERATIONS'////)                                                  00003510
  116 FORMAT('0',' THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AN00003520
     1D ROTATIONS FOLLOW FOR THETA =',E15.6///)                         00003530
  117 FORMAT (/' STATION      N S              N THETA          N STHETA  00003540
     1       Q S              M S              M THETA          M STHETA'//)00003550
  118 FORMAT(1X,I3,3X,7E16.4)                                           00003560
 1181 FORMAT(1X,I3,3X,3E16.4,'  NOT COMPUTED  ',3E16.4)                 00003561
  151 FORMAT('1',' THE TIME STEP NUMBER IS ',I4,'      THE TIME IS ',F500003570
     1.2,' OR ',E9.3,' SECONDS      THE SOLUTION CONVERGED IN ',I2,' ITE00003580
     3RATIONS'////)                                                     00003590
  217 FORMAT(//' STATION       U                V                W       00003600
     1       PHI S            PHI THETA       PHI'/)                   00003610
  218 FORMAT(1X,I3,3X,6E16.4)                                           00003620
  749 FORMAT('1',40X,' MODAL OUTPUT FOR MODE N = ',I3,' FOLLOWS')       00003630
C*******************************************************************00003640
      RETURN                                                            00003650
      END                                                               00003660
```



```
      SUBROUTINE PLOT2(NTH)                                             00000010
C*******************************************************************00000020
C     THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER    *00000030
C     SPECIFIED) OUTPUT QUANTITIES                                     *00000040
C*******************************************************************00000050
      IMPLICIT LOGICAL*1 ($)                                            00000060
      COMMON /IBL4/ KMAX,KL                                             00000070
      COMMON /BL19/ TH(36)                                              00000080
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF, 00000090
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,  00000100
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,     00000110
     3                IPHIT,IPHI,$PLOTS,$MODAL                          00000120
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),  00000130
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),  00000140
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),  00000150
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),  00000160
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),00000170
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),    00000180
```

```
     6                    YPHI(100),XSTATN(100)                                00000190
C*****************************************************************************00000191
      NGKMAX=-KMAX                                                             00000200
      IF ($MODAL) GO TO 121                                                    00000210
      IF (INS.EQ.0) GO TO 4                                                    00000220
      WRITE (6,1000)                                                           00000230
      IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)                            00000240
      IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)                          00000250
      WRITE (6,1001)TH(NTH)                                                    00000260
    4 IF (INTH.EQ.0) GO TO 5                                                   00000270
      WRITE (6,1000)                                                           00000280
      IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)                          00000290
      IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)                        00000300
      WRITE (6,1002) TH(NTH)                                                   00000310
    5 IF (INSTH.EQ.0) GO TO 6                                                  00000320
      WRITE (6,1000)                                                           00000330
      IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)                        00000340
      IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)                      00000350
      WRITE (6,1003) TH(NTH)                                                   00000360
    6 IF (IQS.EQ.0) GO TO 7                                                    00000370
      WRITE (6,1000)                                                           00000380
      IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)                            00000390
      IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)                          00000400
      WRITE (6,1004) TH(NTH)                                                   00000410
    7 IF (IMS.EQ.0) GO TO 8                                                    00000420
      WRITE (6,1000)                                                           00000430
      IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)                            00000440
      IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)                          00000450
      WRITE (6,1005) TH(NTH)                                                   00000460
    8 IF (IMTH.EQ.0) GO TO 9                                                   00000470
      WRITE (6,1000)                                                           00000480
      IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)                          00000490
      IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)                        00000500
      WRITE (6,1006) TH(NTH)                                                   00000510
    9 IF (IMSTH.EQ.0) GO TO 1211                                               00000520
      WRITE (6,1000)                                                           00000530
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                        00000540
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)                      00000550
      WRITE (6,1007) TH(NTH)                                                   00000560
 1211 IF (IU.EQ.0) GO TO 10                                                    00000570
      WRITE (6,1000)                                                           00000580
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                              00000590
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                            00000600
      WRITE (6,1010) TH(NTH)                                                   00000610
   10 IF (IV.EQ.0) GO TO 11                                                    00000620
      WRITE (6,1000)                                                           00000630
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                              00000640
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                            00000650
```

```
      WRITE (6,1009) TH(NTH)                                                    00000660
   11 IF (IW.EQ.0) GO TO 12                                                     00000670
      WRITE (6,1030)                                                            00000680
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                               00000690
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                             00000700
      WRITE (6,1008) TH(NTH)                                                    00000710
   12 IF (IPHIS.EQ.0) GO TO 13                                                  00000720
      WRITE (6,1000)                                                            00000730
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                         00000740
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)                       00000750
      WRITE (6,1011) TH(NTH)                                                    00000760
   13 IF (IPHIT.EQ.0) GO TO 14                                                  00000770
      WRITE (6,1000)                                                            00000780
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                         00000790
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)                       00000800
      WRITE (6,1012) TH(NTH)                                                    00000810
   14 IF (IPHI.EQ.0) GO TO 21                                                   00000820
      WRITE (6,1000)                                                            00000830
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)                         00000840
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)                       00000850
      WRITE (6,1013) TH(NTH)                                                    00000860
   21 RETURN                                                                    00000870
  121 IF (INS.EQ.0) GO TO 15                                                    00000880
      WRITE (6,1000)                                                            00000890
      IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)                             00000900
      IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)                           00000910
      WRITE (6,2001)                                                            00000920
   15 IF (INTH.EQ.0) GO TO 16                                                   00000930
      WRITE (6,1000)                                                            00000940
      IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)                           00000950
      IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)                         00000960
      WRITE (6,2002)                                                            00000970
   16 IF (INSTH.EQ.0) GO TO 17                                                  00000980
      WRITE (6,1000)                                                            00000990
      IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)                         00001000
      IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)                       00001010
      WRITE (6,2003)                                                            00001020
   17 IF (IQS.EQ.0) GO TO 18                                                    00001030
      WRITE (6,1000)                                                            00001040
      IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)                             00001050
      IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)                           00001060
      WRITE (6,2004)                                                            00001070
   18 IF (IMS.EQ.0) GO TO 19                                                    00001080
      WRITE (6,1000)                                                            00001090
      IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)                             00001100
      IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)                           00001110
      WRITE (6,2005)                                                            00001120
   19 IF (IMTH.EQ.0) GO TO 22                                                   00001130
```

```
      WRITE (6,1000)                                                    00001140
      IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)                   00001150
      IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)                 00001160
      WRITE (6,2006)                                                    00001170
   22 IF (IMSTH.EQ.0) GO TO 231                                         00001180
      WRITE (6,1000)                                                    00001190
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                 00001200
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)               00001210
      WRITE (6,2007)                                                    00001220
  231 IF (IU.EQ.0) GO TO 23                                             00001230
      WRITE (6,1000)                                                    00001240
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                       00001250
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                     00001260
      WRITE (6,2010)                                                    00001270
   23 IF (IV.EQ.0) GO TO 24                                             00001280
      WRITE (6,1000)                                                    00001290
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                       00001300
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                     00001310
      WRITE (6,2009)                                                    00001320
   24 IF (IW.EQ.0) GO TO 25                                             00001330
      WRITE (6,1000)                                                    00001340
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                       00001350
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                     00001360
      WRITE (6,2008)                                                    00001370
   25 IF (IPHIS.EQ.0) GO TO 26                                          00001380
      WRITE (6,1000)                                                    00001390
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                 00001400
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)               00001410
      WRITE (6,2011)                                                    00001420
   26 IF (IPHIT.EQ.0) GO TO 27                                          00001430
      WRITE (6,1000)                                                    00001440
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                 00001450
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)               00001460
      WRITE (6,2012)                                                    00001470
   27 IF (IPHI.EQ.0) GO TO 28                                           00001480
      WRITE (6,1000)                                                    00001490
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)                 00001500
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)               00001510
      WRITE (6,2013)                                                    00001520
   28 RETURN                                                            00001530
C*******************************************************************   00001540
 1000 FORMAT ('1')                                                      00001550
 1001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THETA =00001560
     1 ',F10.5,'  RADIANS')                                             00001570
 1002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THE00001580
     1TA = ',F10.5,'  RADIANS')                                         00001590
 1003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATN,  MERIDIAN AT T00001600
     1HETA = ',F10.5,'  RADIANS')                                       00001610
```

```
 1004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATN,  MERIDIAN AT THETA = ',00001620
     1F10.5,'  RADIANS')                                                00001630
 1005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATN,  MERIDIAN AT THETA =00001640
     1 ',F10.5,' RADIANS')                                              00001650
 1006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATN,  MERIDIAN AT THETA00001660
     1 = ',F10.5,'  RADIANS')                                           00001670
 1007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATN,  MERIDIAN AT THE00001680
     1TA = ',F10.5,'  RADIANS')                                         00001690
 1008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATN,  MERIDIAN AT THETA = 00001700
     1',F10.5,'  RADIANS')                                             00001710
 1009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATN,  MERIDIAN AT00001720
     1 THETA = ',F10.5,'  RADIANS')                                     00001730
 1010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATN,  MERIDIAN AT THET00001740
     1A = ',F10.5,'  RADIANS')                                          00001750
 1011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATN,  MERIDIAN AT THETA 00001760
     1= ',F10.5,'  RADIANS')                                            00001770
 1012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATN,  MERIDIAN AT T00001780
     1HETA = ',F10.5,'  RADIANS')                                       00001790
 1013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATN,  MERID00001800
     1IAN AT THETA = ',F10.5,'  RADIANS')                               00001810
 2001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATION')                  00001820
 2002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATION')              00001830
 2003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATION')            00001840
 2004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATION')                     00001850
 2005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATION')                  00001860
 2006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATION')                00001870
 2007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATION')              00001880
 2008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATION')                   00001890
 2009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATION')          00001900
 2010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATION')               00001910
 2011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATION')                 00001920
 2012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATION')            00001930
 2013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATION')    00001940
C*****************************************************************    00001950
      END                                                               00001960
```

```
      SUBROUTINE FLYNVY                                                 00000010
      WRITE (6,111)                                                     00000020
      WRITE (6,1)                                                       00000030
      WRITE (6,2)                                                       00000040
      WRITE (6,3)                                                       00000050
      WRITE (6,4)                                                       00000060
      WRITE (6,5)                                                       00000070
      WRITE (6,6)                                                       00000080
      WRITE (6,7)                                                       00000090
      WRITE (6,8)                                                       00000100
      WRITE (6,9)                                                       00000110
      WRITE (6,10)                                                      00000120
      WRITE (6,11)                                                      00000130
      WRITE (6,12)                                                      00000140
      WRITE (6,13)                                                      00000150
      WRITE (6,14)                                                      00000160
      WRITE (6,15)                                                      00000170
      WRITE (6,16)                                                      00000180
      WRITE (6,17)                                                      00000190
      WRITE (6,18)                                                      00000200
      WRITE (6,19)                                                      00000210
      WRITE (6,20)                                                      00000220
      WRITE (6,21)                                                      00000230
      WRITE (6,22)                                                      00000240
      WRITE (6,23)                                                      00000250
      WRITE (6,24)                                                      00000260
      WRITE (6,25)                                                      00000270
      WRITE (6,26)                                                      00000280
      WRITE (6,27)                                                      00000290
      WRITE (6,28)                                                      00000300
      WRITE (6,29)                                                      00000310
      WRITE (6,30)                                                      00000320
      WRITE (6,31)                                                      00000330
      WRITE (6,32)                                                      00000340
      WRITE (6,33)                                                      00000350
      WRITE (6,34)                                                      00000360
      WRITE (6,35)                                                      00000370
      WRITE (6,36)                                                      00000380
      WRITE (6,37)                                                      00000390
      WRITE (6,38)                                                      00000400
      WRITE (6,39)                                                      00000410
      WRITE (6,40)                                                      00000420
  111 FORMAT ('1')                                                      00000430
    1 FORMAT (' ',T11,'                                                 00000440
     1       XXX')                                                      00000450
    2 FORMAT (' ',T11,'                                                 00000460
     1       XXXXX')                                                    00000470
    3 FORMAT (' ',T11,'                                                 00000480
```

```
     1    XXX   XXX')                                                                                                  00000490
     4 FORMAT (' ',T11,'                                                                                               00000500
     1    XXX   XXX')                                                                                                  00000510
     5 FORMAT (' ',T11,'                                                                                               00000520
     1      X@XXX')                                                                                                    00000530
     6 FORMAT (' ',T11,'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                                                00000540
     1XXXXXXX@XXXXXXXXXX     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00000550
     2XX')                                                                                                             00000560
     7 FORMAT (' ',T11,'   XXXXXXXXOOXXXXXXXOOXXXXXXXOOXXXXXXXXXOX                                                    00000570
     1XXXXXXX@XXXXXXXXXX      XOXXXXXXXXXOOXXXXXXXOOXXXXXXXOOXXXXXXXX'00000580
     2)                                                                                                                00000590
     8 FORMAT (' ',T11,'        XXXXXXOOXXXXXXXOOXXXXXXXOOXXXXXXOX                                                     00000600
     1       @XXX                XOXXXXXXOOXXXXXXXOOXXXXXXOOXXXXXX')                                                  00000610
     9 FORMAT (' ',T11,'          XXXXOOXXXXXXOOXXXXXXXXOOXXXOXX                                                       00000620
     1      XXXXXXXX             XXOXXXOOXXXXXXXOOXXXXXXOOXXXX')                                                      00000630
    10 FORMAT (' ',T11,'                 XXXXXXXXXXXOOXXXXXXXOOXXXXXXXXXXX00000640
     1XXXXXOOOOOOOXXXXXXXXXXXXXXXXOOXXXXXXOOXXXXXXXXXXX')                                                            00000650
    11 FORMAT (' ',T11,'                          XXXXXXXXXXOOXXXXXXOOXXXXXXXOOOOOOO660
     1OOOOOOOOOOOOOOOOOOOOXXXXXXOOXXXXXXOOXXXXXXXXXX')                                                               00000670
    12 FORMAT (' ',T11,'                                XXXXXXXXOOXXXXXOOXXXXXOOOOOOOO680
     1OOOOOOOOOOOOOOOOOOOXXXXXOOXXXXXOOXXXXXXXX')                                                                    00000690
    13 FORMAT (' ',T11,'                                      XXXXXXXOOXXXXOOXXXXOOOOOOO700
     1OOOOOOOOOOOOOOOOOOOXXXXOOXXXXOOXXXXXXX')                                                                       00000710
    14 FORMAT (' ',T11,'                                           XXXXOOXXXOOXXXOOOOO720
     1OX X X X X X X XOOXXOOXXXOOXXXX')                                                                              00000730
    15 FORMAT (' ',T11,'                                               XXOXXXXOOXOOOO740
     1OO X X X X X X OOXOOXXXXOXX')                                                                                  00000750
    16 FORMAT (' ',T11,'                                               OOOOOXXXXXOOOOO760
     1OO X X X X X X OOXXXXXOOOOO')                                                                                  00000770
    17 FORMAT (' ',T11,'                                                 OOO       OOOOO780
     1XOOX X X X X X XOOX      OOO')                                                                                 00000790
    18 FORMAT (' ',T11,'                                                  OOO      OOOOO800
     1 XOO X X X X X OOX      OOO')                                                                                  00000810
    19 FORMAT (' ',T11,'                                                   OOO     OOOOO820
     1  XOOX X X X XOOX      OOO')                                                                                  00000830
    20 FORMAT (' ',T11,'                                                    OOO    OOOOO840
     1  XOO X X X OOX      OOO')                                                                                    00000850
    21 FORMAT (' ',T11,'                                                     OOOO  OOOOO860
     1     XOO X OOX      OOOO')                                                                                   00000870
    22 FORMAT (' ',T11,'                                                      OOOOOOOOO880
     1      XOOOOOX      OOOO')                                                                                    00000890
    23 FORMAT (' ',T11,'                                                       OOOOOOOOO900
     1OOO  OOOXOXOOO  OOOOO')                                                                                       00000910
    24 FORMAT (' ',T11,'                                                                                            00000920
     1 OOOOOOOOOOOOOOO')                                                                                            00000930
    25 FORMAT (' ',T11,'                                                                                            00000940
     1   OOOOOOOOOOO')                                                                                              00000950
    26 FORMAT (' ',T11,'                                                                                            00000960
```

```
     1      OOOOOOOO')                                                                                00000970
    27 FORMAT (' ',T11,'                                                                             00000980
     1        OOO')                                                                                  00000990
    28 FORMAT (' ',/////////)                                                                        00001000
    29 FORMAT (' ',T11,'     FFFFFFFFFF  LLL        YYY          YYY                                 00001010
     1          NNN        NNN         A    VVV       VVV  YYY          YYY                       00001020
     2')                                                                                             00001030
    30 FORMAT (' ',T11,'     FFFFFFFFFF  LLL        YYY        YYY                                   00001040
     1          NNNN       NNN        AAA   VVV       VVV   YYY        YYY'                       00001050
     2)                                                                                              00001060
    31 FORMAT (' ',T11,'     FFF         LLL          YYY     YYY                                   00001070
     1          NNNNN      NNN       AAAAA   VVV     VVV     YYY     YYY')                        00001080
    32 FORMAT (' ',T11,'     FFF         LLL            YYY   YYY                                   00001090
     1          NNNNNN     NNN       AAAAA   VVV     VVV       YYY   YYY')                        00001100
    33 FORMAT (' ',T11,'     FFF         LLL              YYY YYY                                   00001110
     1          NNNNNNN    NNN      AAA AAA   VVV   VVV         YYY YYY')                         00001120
    34 FORMAT (' ',T11,'     FFFFFFFF    LLL                YYYYY                                  00001130
     1          NNN NNNN   NNN      AAA AAA   VVV   VVV         YYYYY')                           00001140
    35 FORMAT (' ',T11,'     FFFFFFFF    LLL                YYY                                    00001150
     1          NNN  NNNN NNN     AAA   AAA   VVV VVV          YYY')                             00001160
    36 FORMAT (' ',T11,'     FFF         LLL                YYY                                    00001170
     1          NNN    NNNNNNN     AAAAAAAAA   VVV VVV          YYY')                             00001180
    37 FORMAT (' ',T11,'     FFF         LLL                YYY                                    00001190
     1          NNN     NNNNNN   AAAAAAAAAAA   VVVVV          YYY')                             00001200
    38 FORMAT (' ',T11,'     FFF         LLL                YYY                                    00001210
     1          NNN       NNNNN   AAA     AAA   VVVVV          YYY')                             00001220
    39 FORMAT (' ',T11,'     FFF         LLLLLLLLLLL        YYY                                    00001230
     1          NNN        NNNN  AAA       AAA   VVV          YYY')                             00001240
    40 FORMAT (' ',T11,'     FFF         LLLLLLLLLLL        YYY                                    00001250
     1          NNN         NNN  AAA       AAA    V           YYY')                             00001260
       RETURN                                                                                        00001270
       END                                                                                           00001280
```

```
      SUBROUTINE FORCE(K)                                                    00000010
C*************************************************************************00000020
C       THIS SUBROUTINE COMPUTES THE GEE VECTOR IN EQUATION (28) OF   *00000030
C REF. 3, AND THE X VECTOR IN EQUATION (29A) OF REF. 3 FOR A GIVEN MER-*00000040
C IDIONAL STATION K.  THE VECTOR 'GEES' IS THE NONLINEAR VALUE OF 'GEE'*00000050
C AT STATION 1.                                                        *00000055
C*************************************************************************00000060
      IMPLICIT LOGICAL*1 ($)                                                 00000070
      REAL NU,MT,LAM2,MASS,MAS                                               00000080
      COMMON /IBL1/ MNMAX                                                    00000090
      COMMON /IBL2/ N(10),MNINIT                                             00000100
      COMMON /IBL4/ KMAX,KL                                                  00000110
      COMMON /IBL8/ LSTEP,ITR                                                00000120
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                       00000130
      COMMON /IBL13/ ITRMAX,LSMAX                                            00000140
      COMMON /BL3/  PR(10),PX(10),PT(10)                                     00000150
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),           00000160
     1              ZF3M(4,4,10),ZF4M(4,4,10)                                00000170
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                             00000180
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                                   00000190
      COMMON /BL7/  D1,S1                                                    00000200
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                 00000210
      COMMON /BL9/  FFS(4,10),EL1S(4),GEES(4,10)                             00000220
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                     00000230
      COMMON /BL12/ TDLI,TDEL                                                00000240
      COMMON /BL14/ LAM2,LSD18,LSD1N                                         00000250
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),     00000260
     1              V3(10),W3(10)                                            00000270
      COMMON /BL17/ DEL                                                      00000280
      COMMON /BL24/ DL(4,4,10),DG(4,4,10),DF(4,4,10)                         00000290
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                         00000300
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)      00000310
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),        00000320
     1              BXT2(10),BE2(10)                                         00000330
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),     00000340
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)               00000350
      COMMON /BL31/ DELSQ,EXT1(10)                                           00000360
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                                00000370
      COMMON /BL100/ TEEO,$DYNMC                                             00000380
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                     00000390
      COMMON /BL102/ DELOAD                                                  00000400
      COMMON /BL103/ MASS(200)                                               00000410
      DIMENSION GEE(4)                                                       00000420
C*************************************************************************00000430
      FDIFF(A,B,C)=(-1.5*A+2.*B-.5*C)/DEL                                    00000440
      RS=R(K)                                                                00000450
      RR=1./RS                                                               00000460
      GA=GAM(K)                                                              00000470
```

```
      OX=OMXI(K)                                                                00000480
      OT=OMT(K)                                                                 00000490
      DL2=D1*LAM2                                                               00000500
      CALL BDB(K,BS,DBS,D,DD)                                                   00000510
      CALL PLOAD(K)                                                             00000520
      CALL TLOAD(K)                                                             00000530
      MAS=MASS(K)                                                               00000540
      DO 4 M=1,MNMAX                                                            00000550
      IZ=K+1+(M-1)*KMAX2                                                        00000560
      IK=K+(M-1)*KMAX                                                           00000570
      IK1=IK-1                                                                  00000580
      EN=N(M)                                                                   00000590
      ENR=EN*RR                                                                 00000600
      EMT=MT(M)                                                                 00000610
      GEE(1) =  (-PX(M)+DT(M)-DL2*GA*OX*EMT)*TDEL*ALOAD                         00000620
     1  + MAS*(-5.*ZO(1,IZ )+4.*Z2(1,IZ)-Z3(1,IZ))*TDEL/DELSD                   00000630
      GEE(2) =  (-PT(M)-ENR*TT(M)-DL2*ENR*OT*EMT)*TDEL*ALOAD                    00000640
     1  + MAS*(-5.*ZO(2,IZ )+4.*Z2(2,IZ)-Z3(2,IZ))*TDEL/DELSD                   00000650
      GEE(3) =  (-PR(M)-(OX+OT)*TT(M)-DL2*(GA*DMT(M)-(OX*OT-ENR**2)             00000660
     1*MT(M)))*TDEL*ALOAD                                                       00000670
     1  + MAS*(-5.*ZO(3,IZ )+4.*Z2(3,IZ)-Z3(3,IZ))*TDEL/DELSD                   00000680
    1 GEE(4) =  MT(M)*TDEL*ALOAD                                                00000690
      IF(ITRMAX.EQ.1) GO TO 50                                                  00000700
      IF(K.GT.1) GO TO 6                                                        00000710
      BX2T=BX1(M)                                                               00000720
      BT2T=BT1(M)                                                               00000730
      BXT2T=BXT1(M)                                                             00000740
      BE2T=BE1(M)                                                               00000750
      EXX2T=EXX1(M)                                                             00000760
      ETT2T=ETT1(M)                                                             00000770
      ETX2T=ETX1(M)                                                             00000780
      EXT2T=EXT1(M)                                                             00000790
      EX2T=EX1(M)                                                               00000800
      ET2T=ET1(M)                                                               00000810
      DBX= FDIFF(BX2T,BX2(M),BX3(M))                                            00000820
      DBT= FDIFF(BT2T,BT2(M),BT3(M))                                            00000830
      DBXT= FDIFF(BXT2T,BXT2(M),BXT3(M))                                        00000840
      DBE= FDIFF(BE2T,BE2(M),BE3(M))                                            00000850
      DET= FDIFF(ET2T,ET2(M),ET3(M))                                            00000860
      DEX= FDIFF(EX2T,EX2(M),EX3(M))                                            00000870
      DEXX= FDIFF(EXX2T,EXX2(M),EXX3(M))                                        00000880
      DETX= FDIFF(ETX2T,ETX2(M),ETX3(M))                                        00000890
      GO TO 7                                                                   00000900
    6 IF(K.LT.KMAX) GO TO 8                                                     00000910
      BX2T=BX3(M)                                                               00000920
      BT2T=BT3(M)                                                               00000930
      BXT2T=BXT3(M)                                                             00000940
      BE2T=BE3(M)                                                               00000950
```

```
      EXX2T=EXX3(M)                                                          00000960
      ETT2T=ETT3(M)                                                          00000970
      ETX2T=ETX3(M)                                                          00000980
      EXT2T=EXT3(M)                                                          00000990
      EX2T=EX3(M)                                                            00001000
      ET2T=ET3(M)                                                            00001010
      DBX=-FDIFF(BX2T,BX2(M),BX1(M))                                         00001020
      DBT=-FDIFF(BT2T,BT2(M),BT1(M))                                         00001030
      DBXT=-FDIFF(BXT2T,BXT2(M),BXT1(M))                                     00001040
      DBE=-FDIFF(BE2T,BE2(M),BE1(M))                                         00001050
      DET=-FDIFF(ET2T,ET2(M),ET1(M))                                         00001060
      DEX=-FDIFF(EX2T,EX2(M),EX1(M))                                         00001070
      DEXX=-FDIFF(EXX2T,EXX2(M),EXX1(M))                                     00001080
      DETX=-FDIFF(ETX2T,ETX2(M),ETX1(M))                                     00001090
      GO TO 7                                                                00001100
    8 DBX=TDLI*(BX3(M)-BX1(M))                                               00001110
      DBT=TDLI*(BT3(M)-BT1(M))                                               00001120
      DBE=TDLI*(BE3(M)-BE1(M))                                               00001130
      DBXT=TDLI*(BXT3(M)-BXT1(M))                                            00001140
      DET=TDLI*(ET3(M)-ET1(M))                                               00001150
      DEX=TDLI*(EX3(M)-EX1(M))                                               00001160
      DEXX=TDLI*(EXX3(M)-EXX1(M))                                            00001170
      DETX=TDLI*(ETX3(M)-ETX1(M))                                            00001180
      BX2T=BX2(M)                                                            00001190
      BT2T=BT2(M)                                                            00001200
      BXT2T=BXT2(M)                                                          00001210
      BE2T=BE2(M)                                                            00001220
      EXX2T=EXX2(M)                                                          00001230
      ETT2T=ETT2(M)                                                          00001240
      ETX2T=ETX2(M)                                                          00001250
      EXT2T=EXT2(M)                                                          00001260
      EX2T=EX2(M)                                                            00001270
      ET2T=ET2(M)                                                            00001280
    7 GEE(1)=GEE(1)-OSE*(BS*(DBX+DBE+GA*D1*(BX2T-BT2T)+NU*(DBT+DBE)          00001290
     1          +ENR*D1*BXT2T)+DBS*(BX2T+BE2T+NU*(BT2T+BE2T))-2.*OX*         00001300
     2                (EXX2T+ETX2T)-ENR*(EX2T+ET2T))*TDEL                    00001310
      GEE(2)=GEE(2)+OSE*(BS*(ENR*(BT2T+BE2T+NU*(BX2T+BE2T))-D1*              00001320
     1      (DBXT+2.*GA*BXT2T))-D1*DBS*BXT2T+2.*OT*(ETT2T+EXT2T)            00001330
     2        -(DEX+DET))*TDEL                                              00001340
      GEE(3)=GEE(3)+OSE*(BS*((OX+NU*OT)*(BX2T+BE2T)+(OT+NU*OX)*              00001350
     1        (BT2T+BE2T))+2.*(GA*(EXX2T+ETX2T)+DEXX+DETX+ENR*               00001360
     2        (EXT2T+ETT2T)))*TDEL                                          00001370
   50 IF(K.GT.1) GO TO 10                                                    00001380
      IF(M.GT.1) EL1S(1)=0.0                                                 00001390
      DO 20 I=1,4                                                            00001400
      GEES(I,M)=GEE(I)                                                       00001410
      SUMX=0.                                                                00001420
      DO 21 J=1,4                                                            00001430
```

```
C*************************************************************************00001440
C      FOLLOWING CARD CAUSES A SPECIFIED BOUNDARY CONDITION VALUE TO     *00001450
C      EXISTS ONLY FOR MODE '0'                                          *00001460
C*************************************************************************00001470
      IF (M.NE.1) EL1S(J)=0                                               00001480
   21 SUMX=SUMX+DL(I,J,M)*EL1S(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)     00001490
   20 X(I,IK)=SUMX                                                        00001500
      GO TO 4                                                             00001510
   10 DO 11 I=1,4                                                         00001520
      SUMX=0.                                                             00001530
      DO 12 J=1,4                                                         00001540
   12 SUMX=SUMX+DEE(I,J,IK)*GEE(J)-DST(I,J,IK)*X(J,IK1)                   00001550
   11 X(I,IK)=SUMX                                                        00001560
    4 CONTINUE                                                            00001570
      RETURN                                                              00001580
      END                                                                 00001590
```



```
      SUBROUTINE MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)          00000010
C*************************************************************************00000020
C         THIS SUBROUTINE SOLVES THE MATRIX EQUATION AX=B, WHERE A IS     *00000030
C         A SQUARE COEFFICIENT MATRIX AND B IS A MATRIX OF CONSTANT VEC-  *00000040
C         TORS.                                                           *00000050
C         A(INVERSE) IS ALSO OBTAINED AND THE DETERMINANT OF A IS AVAIL-  *00000060
C         ABLE.                                                           *00000070
C         THE FOLLOWING MUST BE DIMENSIONED IN THE CALLING PROGRAM:       *00000080
C         IPIVOT(N MAX), INDEX(N MAX,2), A(N MAX,N MAX), B(N MAX,N MAX)   *00000090
C         WHERE:                                                          *00000100
C                   A = NAME OF 2-DIMENSIONAL ARRAY TO BE INVERTED        *00000110
C                   N = ORDER OF A -  1<=N<=NMAX                          *00000120
C                   B = NAME OF 2-DIMENSIONAL ARRAY TO BE MULTIPLIED      *00000130
C                         BY A(INVERSE)                                   *00000140
C                   M = NUMBER OF COLUMN VECTORS IN B                     *00000150
C                         NOTE: M = 0 SIGNALS INVERSION ONLY)             *00000160
C                   IPIVOT = TEMPORARY STORAGE BLOCK                      *00000170
C                   INDEX = TEMPORARY STORAGE BLOCK                       *00000180
C                   NMAX = MAXIMUM ORDER OF A (AS DIMENSIONED IN THE      *00000190
C                         CALLING PROGRAM)                                *00000200
```

```
C                  DETERM= VALUE OF DETERMINANT AS GIVEN BELOW          *00000210
CC                 ISCALE = USED IN FORMULA BELOW                       *00000220
C                                                                       *00000230
CCC                DETERMINANT(A) = (10**18)**ISCALE*(DETERM)           *00000240
C                                                                       *00000250
CC                 A(INVERSE) IS STORED IN A                            *00000260
C                  A(INVERSE)*B IS STORED IN B                          *00000270
C***********************************************************************00000280
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)             00000290
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)         00000300
C***********************************************************************00000310
C     INITIALIZATION                                                    00000320
C                                                                       00000330
    5 ISCALE=0                                                          00000340
    6 R1=10.0**18                                                       00000350
    7 R2=1.0/R1                                                         00000360
   10 DETERM=1.0                                                        00000370
   15 DO 20 J=1,N                                                       00000380
   20 IPIVOT(J)=0                                                       00000390
   30 DO 550 I=1,N                                                      00000400
C                                                                       00000410
CC     SEARCH FOR PIVOT ELEMENT                                         00000420
C                                                                       00000430
   40 AMAX=0.0                                                          00000440
   45 DO 105 J=1,N                                                      00000450
   50 IF (IPIVOT(J)-1) 60, 105, 60                                      00000460
   60 DO 100 K=1,N                                                      00000470
   70 IF (IPIVOT(K)-1) 80, 100, 740                                     00000480
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100                              00000490
   85 IROW=J                                                            00000500
   90 ICOLUM=K                                                          00000510
   95 AMAX=A(J,K)                                                       00000520
  100 CONTINUE                                                          00000530
  105 CONTINUE                                                          00000540
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                   00000550
C                                                                       00000560
CC     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                00000570
C                                                                       00000580
  130 IF (IROW-ICOLUM) 140, 260, 140                                    00000590
  140 DETERM=-DETERM                                                    00000600
  150 DO 200 L=1,N                                                      00000610
  160 SWAP=A(IROW,L)                                                    00000620
  170 A(IROW,L)=A(ICOLUM,L)                                             00000630
  200 A(ICOLUM,L)=SWAP                                                  00000640
  205 IF(M) 260, 260, 210                                               00000650
  210 DO 250 L=1, M                                                     00000660
  220 SWAP=B(IROW,L)                                                    00000670
  230 B(IROW,L)=B(ICOLUM,L)                                             00000680
```

```
  250 B(ICOLUM,L)=SWAP                                                  00000690
  260 INDEX(I,1)=IROW                                                   00000700
  270 INDEX(I,2)=ICOLUM                                                 00000710
  310 PIVOT=A(ICOLUM,ICOLUM)                                            00000720
C                                                                       00000730
C     SCALE THE DETERMINANT                                             00000740
C                                                                       00000750
 1000 PIVOTI=PIVOT                                                      00000760
 1005 IF(ABS(DETERM)-R1)1030,1010,1010                                  00000770
 1010 DETERM=DETERM/R1                                                  00000780
      ISCALE=ISCALE+1                                                   00000790
      IF(ABS(DETERM)-R1)1060,1020,1020                                  00000800
 1020 DETERM=DETERM/R1                                                  00000810
      ISCALE=ISCALE+1                                                   00000820
      GO TO 1060                                                        00000830
 1030 IF(ABS(DETERM)-R2)1040,1040,1060                                  00000840
 1040 DETERM=DETERM*R1                                                  00000850
      ISCALE=ISCALE-1                                                   00000860
      IF(ABS(DETERM)-R2)1050,1050,1060                                  00000870
 1050 DETERM=DETERM*R1                                                  00000880
      ISCALE=ISCALE-1                                                   00000890
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070                                  00000900
 1070 PIVOTI=PIVOTI/R1                                                  00000910
      ISCALE=ISCALE+1                                                   00000920
      IF(ABS(PIVOTI)-R1)320,1080,1080                                   00000930
 1080 PIVOTI=PIVOTI/R1                                                  00000940
      ISCALE=ISCALE+1                                                   00000950
      GO TO 320                                                         00000960
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320                                   00000970
 2000 PIVOTI=PIVOTI*R1                                                  00000980
      ISCALE=ISCALE-1                                                   00000990
      IF(ABS(PIVOTI)-R2)2010,2010,320                                   00001000
 2010 PIVOTI=PIVOTI*R1                                                  00001010
      ISCALE=ISCALE-1                                                   00001020
  320 DETERM=DETERM*PIVOTI                                              00001030
C                                                                       00001040
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                                 00001050
C                                                                       00001060
  330 A(ICOLUM,ICOLUM)=1.0                                              00001070
  340 DO 350 L=1,N                                                      00001080
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT                                     00001090
  355 IF(M) 380, 380, 360                                               00001100
  360 DO 370 L=1,M                                                      00001110
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT                                     00001120
C                                                                       00001130
C     REDUCE NON-PIVOT ROWS                                             00001140
C                                                                       00001150
  380 DO 550 L1=1,N                                                     00001160
```

```
  390 IF(L1-ICOLUM) 400, 550, 400                                        00001170
  400 T=A(L1,ICOLUM)                                                     00001180
  420 A(L1,ICOLUM)=0.0                                                   00001190
  430 DO 450 L=1,N                                                       00001200
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                      00001210
  455 IF(M) 550, 550, 460                                                00001220
  460 DO 500 L=1,M                                                       00001230
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                      00001240
  550 CONTINUE                                                           00001250
C                                                                        00001260
C     INTERCHANGE COLUMNS                                                00001270
C                                                                        00001280
  600 DO 710 I=1,N                                                       00001290
  610 L=N+1-I                                                            00001300
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                           00001310
  630 JROW=INDEX(L,1)                                                    00001320
  640 JCOLUM=INDEX(L,2)                                                  00001330
  650 DO 705 K=1,N                                                       00001340
  660 SWAP=A(K,JROW)                                                     00001350
  670 A(K,JROW)=A(K,JCOLUM)                                              00001360
  700 A(K,JCOLUM)=SWAP                                                   00001370
  705 CONTINUE                                                           00001380
  710 CONTINUE                                                           00001390
  740 RETURN                                                             00001400
      END                                                                00001410
```



```
      SUBROUTINE ABC                                                     00000010
C*****************************************************************     00000020
C         THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE A, BEE, AND C    *00000030
C         MATRICES.                                                     *00000040
C*****************************************************************     00000050
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                               00000060
      COMMON /BL12/ TDLI,TDEL                                            00000070
      COMMON /BL17/ DEL                                                  00000080
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                 00000090
C*****************************************************************     00000100
      D2=2./DEL                                                          00000110
```

```
      DO 1 I=1,4                                                        00000120
      DO 1 J=1,4                                                        00000130
      DEIJ=D2*E(I,J)                                                    00000140
      FIJ=F(I,J)                                                        00000150
      BEE(I,J)=-2.*DEIJ+TDEL*G(I,J)                                     00000160
      C(I,J)=DEIJ-FIJ                                                   00000170
    1 A(I,J)=DEIJ+FIJ                                                   00000180
      RETURN                                                            00000190
      END                                                               00000200
```

```
      SUBROUTINE PANDD(K,MN)                                            00000210
C*****************************************************************    00000220
C         THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE P, P-BAR, AND    *00000230
C         P-HAT MATRICES FOR EACH MEDIDIAN STATION K AND FOURIER MODE MN*00000240
C         THESE MATRICES ARE COMPUTED AND SAVED BECAUSE THEY DO NOT     *00000250
C         CHANGE DURING EITHER THE ITERATION PROCEDURE OR THE LOAD INCRE-00000260
C         MENT PROCEDURE - AS THEY ARE A FUNCTION OF THE SHELL'S INITIAL*00000270
C         GEOMETRY AND STIFFNESS.                                       *00000280
C*****************************************************************    00000290
      COMMON /IBL4/ KMAX,KL                                             00000300
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              00000310
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),      00000320
     1              ZF3M(4,4,10),ZF4M(4,4,10)                           00000330
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                           00000340
      DIMENSION TM(4,4),IPIVOT(4),INDEX(4,2),X2(4)                      00000350
C*****************************************************************    00000360
      IKL=K+KMAX*(MN-1)                                                 00000370
      KLI=IKL-1                                                         00000380
      DO 1 I=1,4                                                        00000390
      DO 1 J=1,4                                                        00000400
      SUM=0.                                                            00000410
      DO 2 L=1,4                                                        00000420
    2 SUM=SUM+C(I,L)*P(L,J,KLI)                                         00000430
    1 TM(I,J)=BEE(I,J)-SUM                                              00000440
      CALL MATINV(TM,4,X2,0,DETERM,IPIVOT,INDEX,4,ISCALE)               00000450
      DO 5 I=1,4                                                        00000460
      DO 5 J=1,4                                                        00000470
```

```
      SUMA=0.                                                          00000480
      SUMC=0.                                                          00000490
      DO 6 L=1,4                                                       00000500
      SUMA=SUMA+TM(I,L)*A(L,J)                                         00000510
    6 SUMC=SUMC+TM(I,L)*C(L,J)                                         00000520
      P(I,J,IKL)=SUMA                                                  00000530
      DEE(I,J,IKL)=TM(I,J)                                             00000540
    5 DST(I,J,IKL)=SUMC                                                00000550
      RETURN                                                           00000560
      END                                                              00000570
```

```
      SUBROUTINE MODES                                                 00000010
C*****************************************************************00000020
C         IN THIS SUBROUTINE, ARRAYS THAT DEFINE THOSE SETS OF INDICES  *00000030
C         THAT COMBINE TO EQUAL EACH VALUE OF N IN THE PROBLEM ARE DETER*00000040
C         MINED.  MODES IS CALLED PRIOR TO THE FIRST ITERATION AND AFTER*00000050
C         EVERY ITERATION UNTIL A SPECIFIED NUMBER OF FOURIER TERMS IS  *00000060
C         REACHED.  EACH FOURIER INDEX IN THE PROBLEM IS SUBTRACTED FROM*00000070
C         ALL OTHER FOURIER INDICES AND THE RESULT IS COMPARED WITH ALL *00000080
C         FOURIER INDICES TO SEE IF THE NEW VALUE EXISTS IN THE PROGRAM.*00000090
C         IF IT DOES, THE LOCATIONS OF THE TWO INDICES THAT MADE THE COM-00000100
C         BINATION ARE STORED IN TWO SPECIAL 2-DIMENSIONAL ARRAYS, ID AND00000110
C         JD.  ONE ARGUMENT OF EACH ARRAY IS THE VALUE OF THE NEW INDEX *00000120
C         AND THE OTHER IS THE NUMBER OF COMBINATIONS OF INDICES THAT   *00000130
C         ALSO GIVE THIS VALUE OF THE INDEX.  IF THERE IS NO INDEX IN THE00000140
C         PROGRAM THAT MATCHES THE NEW ONE, THEN A NEW FOURIER TERM HAS *00000150
C         BEEN GENERATED AND WILL BE CONSIDERED IN THE NEXT ITERATION FOR00000160
C         SOLUTION.                                                     *00000170
C         THE VARIABLE MAXD STORES THE TOTAL NUMBER OF SUCH COMBINATIONS*00000180
C         FOR EACH VALUE OF THE FOURIER INDEX.  IN A SIMILAR MANNER, EACH00000190
C         INDEX IS ADDED TO EVERY OTHER INDEX AND THE SUM COMPARED WITH *00000200
C         ALL INDICES.  THIS RESULT IS STORED IN THE 2-DIMENSIONAL ARRAYS00000210
C         IS AND JS, IN THE SAME MANNER AS WAS DONE FOR THE SUBTRACTION *00000220
C         CASE.  THE VARIABLE MAXS STORES THE TOTAL NUMBER OF SUMMATION *00000230
C         COMBINATIONS FOR EACH VALUE OF THE FOURIER INDEX.             *00000240
C         A SPECIAL ROUTINE HANDLES THE CASES WHERE THE INDEX IS ADDED TO00000250
C         AND SUBTRACTED FROM ITSELF.  THE 2-DIMENSIONAL ARRAY IJS STORES00000260
```

```
C         THE LOCATION OF THE INDEX AND THE VARIABLE MAXSY STORES THE   *00000270
C         TOTAL NUMBER OF SUCH COMBINATIONS.                            *00000280
C         WITH THIS PROCEDURE, THE SERIES OF PRODUCTS THAT MAKE UP THE  *00000290
C         BETA'S AND AETA'S CONTAIN NO ZERO TERM, AND THE SUMMATION IS  *00000300
C         CARRIED OUT IN PHIBET(K) AND TEAETA(K) OVER SPECIFICALLY DE-  *00000310
C         FINED LIMITS.                                                 *00000320
C*************************************************************************00000330
      COMMON /IBL1/ MNMAX                                                00000340
      COMMON /IBL2/ N(10),MNINIT                                         00000350
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),        00000360
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                00000370
      COMMON /IBL9/ MAXM                                                 00000380
      COMMON /IBL11/ ICORFL,IPASS                                        00000390
C*************************************************************************00000400
      IF(MAXM.EQ.1) RETURN                                               00000410
      IF(MNINIT.GT.MAXM) RETURN                                          00000420
      DO 1 MN=1,MNMAXO                                                   00000430
      NMN=N(MN)                                                          00000440
      NNS=MN                                                             00000450
      IF(MNINIT.GT.MN) NNS=MNINIT                                        00000460
      DO 1 MM=NNS,MNMAXO                                                 00000470
      NMM=N(MM)                                                          00000480
      NTEST=IABS(NMN-NMM)                                                00000490
      DO 2 MMFT=1,MNMAX                                                  00000500
      IF(NTEST.EQ.N(MMFT)) GO TO 10                                      00000510
    2 CONTINUE                                                           00000520
      IF(ICORFL.EQ.1) GO TO 1                                            00000530
      MNMAX=MNMAX+1                                                      00000540
      N(MNMAX)=NTEST                                                     00000550
      MMFT=MNMAX                                                         00000560
      IF(MNMAX.EQ.MAXM) ICORFL=1                                         00000570
   10 IF(NMN-NMM) 11,1,12                                                00000580
   11 LOCD=MAXD(MMFT)+1                                                  00000590
      MAXD(MMFT)=LOCD                                                    00000600
      ID(LOCD,MMFT)=MM                                                   00000610
      JD(LOCD,MMFT)=MN                                                   00000620
      GO TO 1                                                            00000630
   12 LOCD=MAXD(MMFT)+1                                                  00000640
      MAXD(MMFT)=LOCD                                                    00000650
      ID(LOCD,MMFT)=MN                                                   00000660
      JD(LOCD,MMFT)=MM                                                   00000670
    1 CONTINUE                                                           00000680
      DO 301 MN=1,MNMAXO                                                 00000690
      NMN=N(MN)                                                          00000700
      NNS=MN                                                             00000710
      IF(MNINIT.GT.MN) NNS=MNINIT                                        00000720
      DO 301 MM=NNS,MNMAXO                                               00000730
      NMM=N(MM)                                                          00000740
```

```
      NTEST=NMN+NMM                                                     00000750
      DO 302 MMFT=1,MNMAX                                               00000760
      IF(NTEST.EQ.N(MMFT)) GO TO 310                                    00000770
  302 CONTINUE                                                          00000780
      IF(ICORFL.EQ.1) GO TO 301                                         00000790
      IF(MNMAX.GE.MAXM) GO TO 301                                       00000800
      MNMAX=MNMAX+1                                                     00000810
      N(MNMAX)=NTEST                                                    00000820
      MMFT=MNMAX                                                        00000830
      IF(MNMAX.GE.MAXM) ICORFL=1                                        00000840
  310 IF(NMN.EQ.NMM) GO TO 360                                          00000850
      LOCS=MAXS(MMFT)+1                                                 00000860
      MAXS(MMFT)=LOCS                                                   00000870
      IS(LOCS,MMFT)=MN                                                  00000880
      JS(LOCS,MMFT)=MM                                                  00000890
      GO TO 301                                                         00000900
  360 IF(NMN.EQ.0) GO TO 301                                            00000910
      MAXSY(MMFT)=1                                                     00000920
      IJS(MMFT)=MN                                                      00000930
  301 CONTINUE                                                          00000940
      MNINIT=MNMAXO+1                                                   00000950
      IF(ICORFL.GT.0) IPASS=IPASS+1                                     00000960
      IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRX                    00000970
      RETURN                                                            00000980
      END                                                               00000990
```



```
      SUBROUTINE HJ(K,MN)                                               00000010
C*****************************************************************    00000020
C         THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE H AND JAY      *00000030
C         MATRICES FOR BOTH BOUNDARIES OF THE SHELL                   *00000040
C*****************************************************************    00000050
      REAL L2,LAM2,LSD1N,      LSD18,JAY,NU                             00000060
      COMMON /IBL2/ N(10),MNINIT                                        00000070
      COMMON /IBL4/ KMAX,KL                                             00000080
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            00000090
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                00000100
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    00000110
```

```
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),00000120
     1              V3(10),W3(10)                                        00000130
      COMMON /BL17/ DEL                                                 00000140
      COMMON /BL20/ DEOMX(200)                                          00000150
      COMMON /BL23/ JAY(4,4),H(4,4)                                     00000160
      EQUIVALENCE(L2,LAM2)                                              00000170
C*****************************************************************     00000180
      CALL BDB(K,B,DB,D,DD)                                             00000190
      YAH=1.                                                            00000200
      IF(K.EQ.1.OR.K.EQ.KMAX)YAH=2.                                     00000210
      D1=(1.-NU)                                                        00000220
      GA=GAM(K)                                                         00000230
      OX=OMXI(K)                                                        00000240
      RA=R(K)                                                           00000250
      EN=N(MN)                                                          00000260
      ENR=EN/RA                                                         00000270
      REG=0.                                                            00000280
      IF(YAH.EQ.2.) REG=1.                                              00000290
      OT=OMT(K)                                                         00000300
      OXT=3.*OMXI(K)-OMT(K)                                             00000310
      OTX=3.*OMT(K)-OMXI(K)                                             00000320
      DL=D*L2*D1*ENR                                                    00000330
      H(1,1)=B                                                          00000340
      H(1,2)=0.                                                         00000350
      H(1,3)=0.                                                         00000360
      H(1,4)=0.                                                         00000370
      H(2,1)=0.                                                         00000380
      H(2,2)=8*D1/2.+L2*D*D1/8.*OTX**2*REG                              00000390
      H(2,3)=DL/2.*OTX*REG                                              00000400
      H(2,4)=0.                                                         00000410
      H(3,1)=0.                                                         00000420
      H(3,2)=DL*OTX*YAH/4.                                              00000430
      ENR2=ENR**2                                                       00000440
      H(3,3)=L2*D*D1*(YAH*ENR2+(1.+NU)*GA**2)                           00000450
      GA2=GA**2                                                         00000460
      H(3,4)=L2                                                         00000470
      H(4,1)=0.                                                         00000480
      H(4,2)=0.                                                         00000490
      H(4,3)=-1.                                                        00000500
      H(4,4)=0.                                                         00000510
      JAY(1,1)=NU*GA*B                                                  00000520
      JAY(1,2)=NU*B*ENR                                                 00000530
      JAY(1,3)=B*(OX+NU*OT)                                             00000540
      JAY(1,4)=0.                                                       00000550
      JAY(2,1)=-B*D1*ENR/2.-DL/8.*OXT*OTX*REG                           00000560
      JAY(2,2)=-GA*H(2,2)                                               00000570
      JAY(2,3)=-GA*H(2,3)                                               00000580
      JAY(2,4)=0.                                                       00000590
```

```
      JAY(3,1)=-L2*D*D1*((1.+NU)*GA2*OX+ENR2/4.*OXT*YAH)                 00000600
      JAY(3,2)=-GA*DL/2.*(     2.*OT*(1.+NU)+OTX/2.*YAH)                 00000610
      JAY(3,3)=-L2*D*D1*(1.+NU+YAH)*GA*ENR2                              00000620
      JAY(3,4)=L2*D1*GA                                                  00000630
      JAY(4,1)=OX                                                        00000640
      JAY(4,2)=0.                                                        00000650
      JAY(4,3)=0.                                                        00000660
      JAY(4,4)=0.                                                        00000670
      DO 1 I=1,4                                                         00000680
      DO 1 J=1,4                                                         00000690
    1 H(I,J)=H(I,J)/2./DEL                                               00000700
      RETURN                                                             00000710
      END                                                                00000720
```



```
      SUBROUTINE POLE(K)                                                 00000010
C*************************************************************************00000020
C        THIS SUBROUTINE PRINTS THE SOLUTION AT AN INITIAL AND A FINAL *00000030
C        POLE.                                                         *00000040
C*************************************************************************00000050
      IMPLICIT LOGICAL*1 ($)                                             00000060
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                  00000070
      COMMON /IBL2/ N(10),MNINIT                                         00000080
      COMMON /IBL3/ M0,M1,M2,M3                                          00000090
      COMMON /IBL4/ KMAX,KL                                              00000100
      COMMON /IBL5/ IBCINL,IBCFNL                                        00000110
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),        00000120
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                00000130
      COMMON /IBL8/ LSTEP,ITR                                            00000140
      COMMON /IBL10/ IFREQ,NTHMAX                                        00000150
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                   00000160
      COMMON /BL4/ P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),        00000170
     1             ZF3M(4,4,10),ZF4M(4,4,10)                             00000180
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                         00000190
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                               00000200
      COMMON /BL7/  D1,S1                                                00000210
      COMMON /BL8/  R(200),GAM(200),OMT(200)                             00000220
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                            00000230
```

```
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                00000240
      COMMON /BL12/ TDLI,TDEL                                           00000250
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    00000260
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),00000270
     1              V3(10),W3(10)                                       00000280
      COMMON /BL17/ DEL                                                 00000290
      COMMON /BL19/ TH(36)                                              00000300
      COMMON /BL20/ DEOMX(200)                                          00000310
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    00000320
      COMMON /BL31/ DELSQ,EXT1(10)                                      00000330
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                 00000340
      COMMON /BL100/ TEEO,$DYNMC                                        00000350
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                00000360
      COMMON /BL102/ DELOAD                                             00000370
      COMMON /BL103/ MASS(200)                                          00000380
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),     00000390
     1               QS(10)                                             00000400
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                             00000410
C*****************************************************************     00000420
      CALL BDB(K,BS,DB,DS,DD)                                           00000430
      IF(K.EQ.KMAX) GO TO 301                                           00000440
      DO 202 MN=1,MNMAXO                                                00000450
      U1(MN)=U2(MN)                                                     00000460
      V1(MN)=V2(MN)                                                     00000470
      W1(MN)=W2(MN)                                                     00000480
      I3=3+(MN-1)*KMAX2                                                 00000490
      I2=I3-1                                                           00000500
      U2(MN)=Z(1,I3)                                                    00000510
      V2(MN)=Z(2,I3)                                                    00000520
      W2(MN)=Z(3,I3)                                                    00000530
      PHIX(MN)=0.                                                       00000540
      PHIT(MN)=0.0                                                      00000550
      PHI(MN)=0.                                                        00000560
      MX(MN)=Z(4,I2)*ABZ3                                               00000570
      MTH(MN)=0.                                                        00000580
      MXT(MN)=0.                                                        00000590
      QS(MN)=0.                                                         00000600
      TX(MN)=0.                                                         00000610
      TTH(MN)=0.                                                        00000620
  202 TXT(MN)=0.                                                        00000630
      IF(M1.EQ.0) GO TO 203                                             00000640
      CALL INLPOL                                                       00000650
      PHIX(M1)=PHEE*ABZO                                                00000660
      PHIT(M1)=-PHEE*ABZO                                               00000670
      QS(M1)=0.                                                         00000680
C     NOTE: CARDS SERIALIZED #00000690 THRU #00000800 HAVE BEEN REMOVED.0000068A
      IF(MO.EQ.0) GO TO 204                                             00000810
       TX(MO) =TO*ABZ                                                   00000820
```

```
      TTH(M0) =T0*ABZ                                                          00000830
      MTH(M0) =MX(M0)                                                          00000840
  234 IF(M2.EQ.0) GO TO 205                                                    00000850
       TX(M2) = T2*ABZ                                                         00000860
      TTH(M2) =-T2*ABZ                                                         00000870
      TXT(M2)=-T2*ABZ                                                          00000880
      MTH(M2) =-MX(M2)                                                         00000890
      MXT(M2) =MTH(M2)                                                         00000900
      GO TO 205                                                                00000910
  203 IF(M0.EQ.0) GO TO 206                                                    00000920
      I3=3+(M0-1)*KMAX2                                                        00000930
      I4=I3+1                                                                  00000940
      CALL TLOAD(1)                                                            00000950
      TX(M0)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ         00000960
     1       -TT(M0)*ABZ*ALOAD                                                 00000970
      TTH(M0)=TX(M0)                                                           00000980
      MTH(M0)=MX(M0)                                                           00000990
  206 IF(M2.EQ.0) GO TO 205                                                    00001000
      I3=3+(M2-1)*KMAX2                                                        00001010
      I4=I3+1                                                                  00001020
      TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL                                 00001030
       TX(M2) = TX(M2)*ABZ                                                     00001040
      TTH(M2) =-TX(M2)                                                         00001050
      TXT(M2) =-TX(M2)                                                         00001060
      MTH(M2) =-MX(M2)                                                         00001070
      MXT(M2) =-MX(M2)                                                         00001080
  205 RETURN                                                                   00001090
  301 CONTINUE                                                                 00001100
      DO 302 MN=1,MNMAXO                                                       00001110
      U1(MN)=U2(MN)                                                            00001120
      V1(MN)=V2(MN)                                                            00001130
      W1(MN)=W2(MN)                                                            00001140
      PHIX(MN)=0.                                                              00001150
      PHIT(MN)=0.                                                              00001160
      PHI(MN)=0.                                                               00001170
      IK=KMAX1+(MN-1)*KMAX2                                                    00001180
      MX(MN)=Z(4,IK)*ABZ3                                                      00001190
      MTH(MN)=0.                                                               00001200
      MXT(MN)=0.                                                               00001210
      QS(MN)=0.                                                                00001220
      TX(MN)=0.                                                                00001230
      TTH(MN)=0.                                                               00001240
  302 TXT(MN)=0.                                                               00001250
      IF(M1.EQ.0) GO TO 303                                                    00001260
      CALL FNLPOL                                                              00001270
      PHIX(M1)=PHEE*ABZO                                                       00001280
      PHIT(M1)= PHEE*ABZO                                                      00001290
      QS(M1)=0.                                                                00001300
```

```
C     NOTE: CARDS SERIALIZED #00001310 THRU #00001430 HAVE BEEN REMOVED.0000130A
      IF(M0.EQ.0) GO TO 304                                              00001440
       TX(M0) =T0*ABZ                                                    00001450
      TTH(M0) =T0*ABZ                                                    00001460
      MTH(M0) =MX(M0)                                                    00001470
  304 IF(M2.EQ.0) GO TO 305                                              00001480
       TX(M2) = T2*ABZ                                                   00001490
      TTH(M2) =-T2*ABZ                                                   00001500
      TXT(M2) = T2*ABZ                                                   00001510
      MTH(M2) =-MX(M2)                                                   00001520
      MXT(M2) =MX(M2)                                                    00001530
      GO TO 305                                                          00001540
  303 IF(M0.EQ.0) GO TO 306                                              00001550
      IKM=KMAX+(M0-1)*KMAX2                                              00001560
      IM1=IKM-1                                                          00001570
      CALL TLOAD(KMAX)                                                   00001580
      TX(M0)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+OMXI(KMAX)*Z(3,IKM+1)00001590
     1          )*ABZ-TT(M0)*ABZ*ALOAD                                   00001600
      TTH(M0) =TX(M0)                                                    00001610
      MTH(M0) =MX(M0)                                                    00001620
  306 IF(M2.EQ.0) GO TO 305                                              00001630
      IKM=KMAX+(M2-1)*KMAX2                                              00001640
      IM1=IKM-1                                                          00001650
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL                        00001660
      TX(M2)=TX(M2)*ABZ                                                  00001670
      TTH(M2) =-TX(M2)                                                   00001680
      TXT(M2) = TX(M2)                                                   00001690
      MTH(M2) =-MX(M2)                                                   00001700
      MXT(M2) = MX(M2)                                                   00001710
  305 RETURN                                                             00001720
      END                                                                00001730
```



```
      SUBROUTINE PLOTIT(X,Y,NN,MODCUR)                                   00000010
C*****************************************************************      00000020
C     THIS SUBROUTINE AND THE THREE THAT FOLLOW IT COMPRISE THE SELF-  *00000030
C CONTAINED PLOTTING CAPABILITY OF THE PROGRAM, SATANS.  THEY RECEIVE  *00000040
C THE DATA TO BE PLOTTED, ROUND IT, SCALE IT AND DRAW IT ON THE HIGH-  *00000050
```

```
C SPEED LINE PRINTER.                                                    *00000060
C*************************************************************************00000070
      DIMENSION X( 1), Y( 1), RANGE(4)                                   00000080
      EQUIVALENCE (RANGE(1),XMAX), (RANGE(2),XMIN),(RANGE(3),YMAX),      00000090
     1           (RANGE(4),YMIN)                                         00000100
C                                                                        00000110
      KN=IABS(NN)                                                        00000120
      IF(MODCUR.GT.1) GO TO 5                                            00000130
C*************************************************************************00000140
C     FIND MAX & MIN FOR SCALE COMPUTATIONS                              *00000150
C*************************************************************************00000160
      XMAX=-1.E20                                                        00000170
      XMIN=1.E20                                                         00000180
      YMAX=-1.E20                                                        00000190
      YMIN=1.E20                                                         00000200
      DO 1 I=1,KN                                                        00000210
      IF(X(I).LT.XMAX) GO TO 6                                           00000220
      XMAX=X(I)                                                          00000230
    6 IF(X(I).GT.XMIN) GO TO 7                                           00000240
      XMIN=X(I)                                                          00000250
    7 IF(Y(I).LT.YMAX) GO TO 8                                           00000260
      YMAX=Y(I)                                                          00000270
    8 IF(Y(I).GT.YMIN) GO TO 1                                           00000280
      YMIN=Y(I)                                                          00000290
    1 CONTINUE                                                           00000300
C*************************************************************************00000310
C     IF NOT AUTOSCALE GO TO CALL DRAWIT                                 *00000320
C*************************************************************************00000330
      IF(NN.GT.0) GO TO 5                                                00000340
C*************************************************************************00000350
C     COMPUTE X-SCALE & NEW XMAX AND XMIN                                *00000360
C*************************************************************************00000370
      CALL SCALIT(XMAX,XMIN,4)                                           00000380
C*************************************************************************00000390
C     COMPUTE Y-SCALE & NEW YMAX AND YMIN                                *00000400
C*************************************************************************00000410
      CALL SCALIT(YMAX,YMIN,6)                                           00000420
C*************************************************************************00000430
C     PLOT CURVE                                                         *00000440
C*************************************************************************00000450
    5 CALL DRAWIT(X,Y,KN,RANGE,1,MODCUR)                                 00000460
      IF(MODCUR. EQ.1.OR.MODCUR.EQ.2) RETURN                             00000470
C*************************************************************************00000480
C     PRINT SCALES WHEN LAST CURVE PLOTTED                               *00000490
C*************************************************************************00000500
      XS=(XMAX-XMIN)/80.                                                 00000510
      YS=(YMAX-YMIN)/60.                                                 00000520
      WRITE(6,100) XS,YS                                                 00000530
```

```
  100 FORMAT(  15X,'X-SCALE:  "*"=',E10.3,' UNITS'//                        00000540
     1         15X,'Y-SCALE:  "*"=',E10.3,' UNITS')                         00000550
       RETURN                                                              00000560
       END                                                                 00000570
C*************************************************************************00000580

       SUBROUTINE SCALIT(XMAX,XMIN,IDIV)                                   00000590
       DIV=IDIV                                                            00000600
C*************************************************************************00000610
C      ROUND MAXIMUM TO NEXT HIGHEST 2 SIG FIGS                           *00000620
C*************************************************************************00000630
       XMAX=AMAX1(0.,XMAX)                                                 00000640
       CALL ROUND(XMAX,IMX,FMX)                                            00000650
       IMX=IMX-1                                                           00000660
       FMX=FMX*10.                                                         00000670
     3 XMX=FMX*10.**IMX                                                    00000680
       IF(XMX.GE.XMAX) GO TO 2                                             00000690
       FMX=FMX+1.                                                          00000700
       IMM=FMX                                                             00000710
       FMX=IMM                                                             00000720
       GO TO 3                                                             00000730
C*************************************************************************00000740
C      ROUND MINIMUM TO NEXT LOWEST 2 SIG FIGS                            *00000750
C*************************************************************************00000760
     2 XMIN=AMIN1(0.,XMIN)                                                 00000770
       CALL ROUND(XMIN,IMN,FMN)                                            00000780
       IMN=IMN-1                                                           00000790
       FMN=FMN*10.                                                         00000800
    14 XMN=FMN*10.**IMN                                                    00000810
       IF(XMIN.GE.XMN) GO TO 11                                            00000820
       FMN=FMN-1.                                                          00000830
       IMM=FMN                                                             00000840
       FMN=IMM                                                             00000850
       GO TO 14                                                            00000860
C*************************************************************************00000870
C      ROUND MAX & MIN TO 1. OR .1 IF RANGE LARGE                         *00000880
C*************************************************************************00000890
    11 XSC=XMX-XMN                                                         00000900
       IM=0                                                                00000910
       SM=1.                                                               00000920
     9 IF(XSC/DIV.LE.SM) GO TO 12                                          00000930
       IF(ABS(XMN).LT.SM.AND.ABS(XMN).GT.0.) XMN=SIGN(SM,XMN)              00000940
       IF(ABS(XMX).LT.SM.AND.ABS(XMX).GT.0.) XMX=SIGN(SM,XMX)              00000950
    12 IF(IM.GT.0) GO TO 19                                                00000960
       SM=.1                                                               00000970
       IM=IM+1                                                             00000980
       GO TO 9                                                             00000990
```

```
C*************************************************************************00001000
C      ROUND RANGE (MAX-MIN) TO 2 SIG FIGS                                *00001010
C*************************************************************************00001020
   19 XSC=XMX-XMN                                                          00001030
      CALL ROUND(XSC,IS10,FACTX)                                           00001040
C*************************************************************************00001050
C      FIND FACTOR WHICH IS MULTIPLE OF IDIV                              *00001060
C*************************************************************************00001070
      FACTX=FACTX*10.                                                      00001080
      OFAC=FACTX                                                           00001090
      IS10=IS10-1                                                          00001100
      IFX=FACTX                                                            00001110
      FACTX=IFX                                                            00001120
   20 IF(MOD(IFX,IDIV).EQ.0.AND.FACTX.GE.OFAC) GO TO 10                    00001130
      IFX=IFX+1                                                            00001140
      FACTX=IFX                                                            00001150
      GO TO 20                                                             00001160
   10 IF(IDIV.GT.4) GO TO 15                                               00001170
C*************************************************************************00001180
C      IF X SCALE BETWEEN 8. AND 10., ROUND TO 10.                        *00001190
C*************************************************************************00001200
      FFX=ABS(FACTX/10.)                                                   00001210
      IF(FFX.GT.8..AND.FFX.LT.10.) FFX=10.                                 00001220
      IF(FACT X.LT.0.) FXX=-10.                                            00001230
      FACTX=FFX*10.                                                        00001240
   15 XSC=FACTX*10.**IS10                                                  00001250
C*************************************************************************00001260
C      COMPUTE NEW MAX & MIN FROM ROUNDED SCALE                           *00001270
C*************************************************************************00001280
      IF(XM N*XM X.NE.0.) GO TO 4                                          00001290
      IF(XM N.LT.0.) XMIN=-XSC                                             00001300
      IF(XM X.GT.0.) XMAX=XSC                                              00001310
      RETURN                                                               00001320
    4 XMAX=XSC+XMN                                                         00001330
      XMIN=XMN                                                             00001340
      RETURN                                                               00001350
      END                                                                  00001360
C*************************************************************************00001370

      SUBROUTINE ROUND(ANUM,IS,FACT)                                       00001380
C*************************************************************************00001390
C      EXPRESS ANUM IN SCIENTIFIC NOTATION WHERE                          *00001400
C           ANUM=FACT*10.**IS WHERE FACT IS BETWEEN 1. AND 9.9            *00001410
C*************************************************************************00001420
      IF(ANUM.EQ.0.) GO TO 15                                              00001430
      BNUM=ANUM                                                            00001440
      IF(BNUM.LT.0.) BNUM=-BNUM                                            00001450
```

```
      IS=ALOG(BNUM)*.43429448                                                  00001460
      FACT=BNUM/10.**IS                                                        00001470
C*************************************************************************00001480
C     FIND POWER OF 10                                                      *00001490
C*************************************************************************00001500
      IDD=-3                                                                   00001510
      R2=0                                                                     00001520
      DO 10 II=1,5                                                             00001530
      IDD=IDD+1                                                                00001540
      R1=R2                                                                    00001550
      R2=10.**(IDD+1)                                                          00001560
      IF(FACT.GE.R1.AND.FACT.LT.R2) GO TO 8                                    00001570
   10 CONTINUE                                                                 00001580
    8 FACT=FACT*10.**(-IDD)                                                    00001590
      IS=IS+IDD                                                                00001600
C*************************************************************************00001610
C     ROUND MANTISSA TO 2 SIG FIGS                                          *00001620
C*************************************************************************00001630
      IFAC=FACT*10.+.05                                                        00001640
      FACT=IFAC                                                                00001650
      FACT=FACT/10.                                                            00001660
      IF(FACT.LT.10.) GO TO 20                                                 00001670
C*************************************************************************00001680
C     SET TO 1 IF LESS THAN 10.                                             *00001690
C*************************************************************************00001700
      FACT=1.                                                                  00001710
      IS=IS+1                                                                  00001720
C*************************************************************************00001730
C     IF INPUT NEGATIVE, SET MANTISSA NEGATIVE                              *00001740
C*************************************************************************00001750
   20 IF(ANUM.LT.0.) FACT=-FACT                                                00001760
      RETURN                                                                   00001770
C*************************************************************************00001780
C     SET TO 0. IF 0.                                                       *00001790
C*************************************************************************00001800
   15 FACT=0.                                                                  00001810
      IS=0                                                                     00001820
      RETURN                                                                   00001830
      END                                                                      00001840
C*************************************************************************00001850


      SUBROUTINE DRAWIT(X,Y,NDATA,RANGE,KKZ,MODCUR)                            00001860
      DIMENSION GRID(61,81),XSCALE(5),YSCALE(7)                                00001870
      DIMENSION X(1),Y(1),RANGE(4)                                             00001880
      INTEGER*2 GRID,BLANK,DOT,XCHAR(4)/1H+,1H.,1H*,1HX/                       00001890
      DATA DOT,BLANK/Z4B40,Z4040/                                              00001900
      KDATA=NDATA*KKZ                                                          00001910
```

```
      IF(MODCUR.GT.1) GO TO 444                                          00001920
C*****************************************************************     00001930
C     GRID IS THE MATRIX USED TO PLOT THE POINTS                      *00001940
C*****************************************************************     00001950
      IERR=0                                                             00001960
      XMAX=RANGE(1)                                                      00001970
      XMIN=RANGE(2)                                                      00001980
      YMAX=RANGE(3)                                                      00001990
      YMIN=RANGE(4)                                                      00002000
C*****************************************************************     00002010
C     CHECKING X AND Y POINTS AND PLOTTING THOSE OUT OF RANGE         *00002020
C     AT THE MARGIN                                                   *00002030
C*****************************************************************     00002040
      DO 30 I=1,KDATA,KKZ                                                00002050
      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.OR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)   00002060
     1  IERR=IERR+1                                                      00002070
      IF(X(I).LE.XMAX) GO TO 205                                         00002080
      X  (I)=XMAX                                                        00002090
      GOTO 210                                                           00002100
  205 IF(X(I).GE.XMIN) GO TO 210                                         00002110
      X  (I)=XMIN                                                        00002120
  210 IF(Y(I).LE.YMAX) GO TO 215                                         00002130
      Y  (I)=YMAX                                                        00002140
      GOTO 30                                                            00002150
  215 IF(Y(I).GE.YMIN) GO TO 30                                          00002160
      Y  (I)=YMIN                                                        00002170
   30 CONTINUE                                                           00002180
C*****************************************************************     00002190
C     PLOTTING X AND Y AXIS , IF NECESSARY                            *00002200
C*****************************************************************     00002210
      JERR=0                                                             00002220
      XRANGE=XMAX-XMIN                                                   00002230
      YRANGE=YMAX-YMIN                                                   00002240
      IF (YRANGE.NE.0.)  GO TO 298                                       00002250
      IF(YMIN.EQ.0.) GO TO 889                                           00002260
      YMIN=0.                                                            00002270
      YRANGE=YMAX                                                        00002280
      GO TO 299                                                          00002290
  298 IF (XRANGE.NE.0.)  GO TO 299                                       00002300
      IF(XMIN.EQ.0.) GO TO 887                                           00002310
      XMIN=0.                                                            00002320
      XRANGE=XMAX                                                        00002330
C*****************************************************************     00002340
C     BLANKING OUT MATRIX-(GRID)                                      *00002350
C*****************************************************************     00002360
  299 DO 300 I=1,61                                                      00002370
      DO 301 JJ=1,81                                                     00002380
301   GRID(I,JJ)=BLANK                                                   00002390
```

```
  300 CONTINUE                                                          00002400
      IF(XMAX*XMIN.GE.0.) GO TO 222                                     00002410
      IYAXIS=80.*(-XMIN)/XRANGE+1.5                                     00002420
      DO 40 I=1,61                                                      00002430
   40 GRID(I,IYAXIS)=DOT                                                00002440
  222 IF(YMAX*YMIN.GE.0.) GO TO 333                                     00002450
      IXAXIS=60.*YMAX/YRANGE+1.5                                        00002460
      DO 60 I=1,81                                                      00002470
   60 GRID(IXAXIS,I)=DOT                                                00002480
C*************************************************************************00002490
C     COMPUTE PROPER SCALE NUMBERS                                      *00002500
C*************************************************************************00002510
  333 XINCR=XRANGE/4.                                                   00002520
      YINCR=YRANGE/6.                                                   00002530
      XSCALE(1)=XMAX                                                    00002540
      XSCALE(5)=XMIN                                                    00002550
      DO 80 I=2,4                                                       00002560
      XSCALE(I)=XSCALE(I-1)-XINCR                                       00002570
      IF(ABS(XSCALE(I)).LT.1.E-4) XSCALE(I)=0.                          00002580
   80 CONTINUE                                                          00002590
      YSCALE(1)=YMAX                                                    00002600
      YSCALE(7)=YMIN                                                    00002610
      DO 81 I=2,6                                                       00002620
      YSCALE(I)=YSCALE(I-1)-YINCR                                       00002630
      IF(ABS(YSCALE(I)).LT.1.E-4) YSCALE(I)=0.                          00002640
   81 CONTINUE                                                          00002650
      DO 85 II=1,2                                                      00002660
      JJ=6-II                                                           00002670
      XT=XSCALE(JJ)                                                     00002680
      XSCALE(JJ)=XSCALE(II)                                             00002690
   85 XSCALE(II)=XT                                                     00002700
C*************************************************************************00002710
C*************************************************************************00002720
  444 IF(MODCUR.LT.2) JSET=0                                            00002730
      IF(JERR.GT.0) GO TO 885                                           00002740
      JSET=JSET+1                                                       00002750
      IF(JSET.GT.4) JSET=1                                              00002760
      DO 700 I=1,KDATA,KKZ                                              00002770
      IPTX=60.*(YMAX-Y(I))/YRANGE+1.5                                   00002780
      IPTY=80.*(X(I)-XMIN)/XRANGE+1.5                                   00002790
      IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70                             00002800
      IF(IPTX.LE.0.OR.IPTY.LE.0)GO TO 70                                00002810
      GRID(IPTX,IPTY) = XCHAR(JSET)                                     00002820
      GO TO 700                                                         00002830
   70 IERR=IERR+1                                                       00002840
  700 CONTINUE                                                          00002850
C*************************************************************************00002860
C     OUTPUT SECTION WITH GRAPH                                         *00002870
```

```
C*************************************************************************00002880
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                                 00002890
      AXR=ABS(XRANGE)                                                       00002900
      AYR=ABS(YRANGE)                                                       00002910
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 400                             00002920
      WRITE(6,17) XSCALE                                                    00002930
   17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('+*********'),'+**')       00002940
      GO TO 401                                                             00002950
  400 WRITE(6,117) XSCALE                                                   00002960
  117 FORMAT( 8X,F11.2,4( 9X,F11.2)/15X,'**',8('+*********'),'+**')         00002970
  401 II=1                                                                  00002980
      DO 101 IK=1,61                                                        00002990
      IF(MOD(IK-1,10).NE.0) GO TO 92                                        00003000
      IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404                             00003010
      WRITE(6,18) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)               00003020
   18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                     00003030
      GO TO 405                                                             00003040
  404 WRITE(6,118) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)              00003050
  118 FORMAT(2X,F11.2,'  + ',81A1,' +  ',F11.2)                             00003060
  405 II=II+1                                                               00003070
      GO TO 101                                                             00003080
   92 WRITE(6,19) (GRID(IK,IX),IX=1,81)                                     00003090
   19 FORMAT(15X,'* ',81A1,' *')                                            00003100
  101 CONTINUE                                                              00003110
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 402                             00003120
      WRITE(6,22) XSCALE                                                    00003130
   22 FORMAT(15X,'**',8('+*********'),'+**'/12X,1PE10.3,4(10X,E10.3),//)     00003140
      GO TO 403                                                             00003150
  402 WRITE(6,217) XSCALE                                                   00003160
  217 FORMAT(15X,'**',8('+*********'),'+**'/ 8X,F11.2,4( 9X,F11.2),//)       00003170
  403 IF(IERR.GT.0) WRITE(6,20) IERR                                        00003180
   20  FORMAT(10X 'NUMBER OF POINTS OUT OF RANGE ='  I4)                    00003190
 1000 RETURN                                                                00003200
  889 WRITE(6,888)                                                          00003210
  888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y    00003220
     1 WHEN MODCUR=0 OR 1.')                                                00003230
      JERR=10                                                               00003240
      RETURN                                                                00003250
  887 WRITE(6,886)                                                          00003260
  886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN X    00003270
     1 WHEN MODCUR=0 OR 1.')                                                00003280
      JERR=10                                                               00003290
      RETURN                                                                00003300
  885 WRITE(6,884)                                                          00003310
  884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRI    00003320
     1D PROPERLY SETUP')                                                    00003330
      RETURN                                                                00003340
      END                                                                   00003350
```

```
      SUBROUTINE XANDZ                                                  00000010
C*****************************************************************00000020
C         THIS SUBROUTINE COMPUTES THE X VECTOR USING THE P, PBAR, AND  *00000030
C         P-HAT MATRICES AND SOLVES FOR THE Z VECTOR FOR THE APPLIED AND*00000040
C         PSUEDO LOADS.  THE SUBROUTINES PHIBET(K) AND TEAETA(K) ARE    *00000050
C         CALLED AND THE PREVIOUS SOLUTION FOR Z, OR THE ESTIMATED VALUE*00000060
C         OF Z IS USED TO CALCULATE THE NON-LINEAR BETA AND AETA TERMS. *00000070
C         THE MATRICES FFS AND FLS ARE THE VALUES OF SMALL-F AT THE     *00000080
C         INITIAL AND FINAL EDGES OF THE SHELL.  THE SUBROUTINE FORCE(K)*00000090
C         IS CALLED TO CALCULATE THE LOAD VECTOR GEE, AND THE X VECTOR AT00000100
C         EACH MERIDIAN STATION.  ONCE THE X VECTOR IS OBTAINED FOR ALL *00000110
C         MERIDIAN STATIONS AND THE SOLUTION FOR Z(K+1) GIVEN BY EQUATIONO0000120
C         (31B) OF REF. 3 IS OBTAINED, AND THE SOLUTION FOR Z(I) AT ALL *00000130
C         OTHER MERIDIAN STATIONS DEFINED BY EQUATION (29B) OF REF. 3 IS*00000140
C         OBTAINED.  THE SOLUTION Z AT THE IMAGINARY STATION OFF THE    *00000150
C         INITIAL EDGE OF THE SHELL IS OBTAINED LAST.  THE TEST FOR CON-*00000160
C         VERGENCE OF THE SOLUTION IS MADE AS Z IS COMPUTED.  THE SPE-  *00000170
C         CIAL CONDITIONS FOR COMPUTING Z AT EITHER AN INITIAL POLE OR A*00000180
C         FINAL POLE ARE ALSO  IN THIS ROUTINE.                         *00000190
C*****************************************************************00000200
      IMPLICIT LOGICAL*1 ($)                                            00000210
      REAL NU,LAM2,MT,MASS                                              00000220
      COMMON /IBL1/ MNMAX                                               00000230
      COMMON /IBL3/ MO,M1,M2,M3                                         00000240
      COMMON /IBL4/ KMAX,KL                                             00000250
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000260
      COMMON /IBL6/ KLL                                                 00000270
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       00000280
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               00000290
      COMMON /IBL8/ LSTEP,ITR                                           00000300
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000310
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000320
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              00000330
      COMMON /BL3/  PR(10),PX(10),PT(10)                                00000340
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),      00000350
     1              ZF3M(4,4,10),ZF4M(4,4,10)                           00000360
```

```
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                                00000370
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                                      00000380
      COMMON /BL7/  D1,S1                                                       00000390
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                    00000400
      COMMON /BL9/  FFS(4,10),EL1S(4),GEES(4,10)                                00000410
      COMMON /BL14/ LAM2,LSD18,LSD1N                                            00000420
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),        00000430
     1              V3(10),W3(10)                                               00000440
      COMMON /BL16/ EPS                                                         00000450
      COMMON /BL18/ EL1(4),ELL(4)                                               00000460
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                            00000470
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)         00000480
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),           00000490
     1              BXT2(10),BE2(10)                                            00000500
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),        00000510
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)                  00000520
      COMMON /BL31/ DELSQ,EXT1(10)                                              00000530
      COMMON /BL100/ TEEO,$DYNMC                                                00000540
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                        00000550
      COMMON /BL102/ DELOAD                                                     00000560
      COMMON /BL103/ MASS(200)                                                  00000570
      COMMON /BL104/ ZDOT(4,220)                                                00000580
      DIMENSION ELLS(4),FLS(4),ZT(4),IPIVOT(4),INDEX(4,2)                       00000590
     1,CLO(4,4),CL1(4,4),CL2(4,4)                                               00000600
     2,TZMAX(4,10),ZDD(4)                                                       00000610
      EQUIVALENCE (CLO(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1))            00000620
C*****************************************************************************00000630
      DO 201 I=1,4                                                              00000640
      DO 201 M=1,MNMAX                                                          00000650
      MJ=1+(M-1)*KMAX2                                                          00000660
      TZMAX(I,M)=ABS(Z(I,MJ))                                                   00000670
      DO 201 K=2,KMAX2                                                          00000680
      KM=K+(M-1)*KMAX2                                                          00000690
      AZTST=ABS(Z(I,KM))                                                        00000700
      IF(AZTST.GT.TZMAX(I,M))  TZMAX(I,M)=AZTST                                 00000710
  201 CONTINUE                                                                  00000720
      NCONV=1                                                                   00000730
      IF(ITRMAX.EQ.1) GO TO 66                                                  00000740
      DO 1 M=1,MNMAXO                                                           00000750
      I=1+(KMAX+2)*(M-1)                                                        00000760
      U1(M)=Z(1,I)                                                              00000770
      V1(M)=Z(2,I)                                                              00000780
      W1(M)=Z(3,I)                                                              00000790
      I1=I+1                                                                    00000800
      U2(M)=Z(1,I1)                                                             00000810
      V2(M)=Z(2,I1)                                                             00000820
    1 W2(M)=Z(3,I1)                                                             00000830
      IF(IBCINL.LT.0) GO TO 100                                                 00000840
```

```
      CALL PHIBET(1)                                                    00000850
      DO 2 M=1,MNMAX                                                    00000860
      BX1(M)=BX3(M)                                                     00000870
      BT1(M)=BT3(M)                                                     00000880
      BXT1(M)=BXT3(M)                                                   00000890
    2 BE1(M)=BE3(M)                                                     00000900
      CALL TEAETA(1)                                                    00000910
      DO 3 M=1,MNMAX                                                    00000920
      EXX1(M)=EXX3(M)                                                   00000930
      ETT1(M)=ETT3(M)                                                   00000940
      ETX1(M)=ETX3(M)                                                   00000950
      EXT1(M)=EXT3(M)                                                   00000960
      EX1(M)=EX3(M)                                                     00000970
    3 ET1(M)=ET3(M)                                                     00000980
  102 CALL PHIBET(2)                                                    00000990
      DO 4 M=1,MNMAX                                                    00001000
      BX2(M)=BX3(M)                                                     00001010
      BT2(M)=BT3(M)                                                     00001020
      BXT2(M)=BXT3(M)                                                   00001030
    4 BE2(M)=BE3(M)                                                     00001040
      CALL TEAETA(2)                                                    00001050
      DO 5 M=1,MNMAX                                                    00001060
      EXX2(M)=EXX3(M)                                                   00001070
      ETT2(M)=ETT3(M)                                                   00001080
      ETX2(M)=ETX3(M)                                                   00001090
      EXT2(M)=EXT3(M)                                                   00001100
      EX2(M)=EX3(M)                                                     00001110
    5 ET2(M)=ET3(M)                                                     00001120
      CALL PHIBET(3)                                                    00001130
      CALL TEAETA(3)                                                    00001140
   66 CONTINUE                                                          00001150
      IF(IBCINL.LT.0) GO TO 20                                          00001160
      CALL BDB(1,B1,DB,D,DD)                                            00001170
      GAM1=GAM(1)                                                       00001180
      CALL TLOAD(1)                                                     00001190
      DO 8 M=1,MNMAX                                                    00001200
      IF(ITRMAX.EQ.1) GO TO 67                                          00001210
      FFS(1,M)=-TT(M)*ALOAD+OSE*(BX1(M)+BE1(M)+NU*(BT1(M)+BE1(M)))*B1   00001220
      FFS(2,M)=OSE*(B1   *D1      *BXT1(M)+EX1(M)+ET1(M))               00001230
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD-(EXX1(M)+ETX1(M))*SOE           00001240
      GO TO 8                                                           00001250
   67 FFS(1,M)=-TT(M)*ALOAD                                             00001260
      FFS(2,M)=0.                                                       00001270
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD                                 00001280
    8 FFS(4,M)=0.                                                       00001290
      DO 9 I=1,4                                                        00001300
    9 EL1S(I)=ALOAD*EL1(I)                                              00001310
      CALL FORCE(1)                                                     00001320
```

```
   20 CALL FORCE(2)                                                                  00001330
      DO 10 K=3,KLL                                                                  00001340
      KP=K+1                                                                         00001350
      IF(ITRMAX.EQ.1) GO TO 10                                                       00001360
      CALL UPDATE                                                                    00001370
      CALL PHIBET(KP)                                                                00001380
      CALL TEAETA(KP)                                                                00001390
   10 CALL FORCE(K)                                                                  00001400
      IF(ITRMAX.NE.1) CALL UPDATE                                                    00001410
      IF(IBCFNL.LT.0) GO TO 120                                                      00001420
      IF(ITRMAX.EQ.1) GO TO 11                                                       00001430
      CALL PHIBET(KMAX)                                                              00001440
      CALL TEAETA(KMAX)                                                              00001450
   11 CALL FORCE(KL)                                                                 00001460
      CALL FORCE(KMAX)                                                               00001470
      DO 12 I=1,4                                                                    00001480
   12 ELLS(I)=ALOAD*ELL(I)                                                           00001490
      CALL BDB(KMAX,BL,DB,D,DD)                                                      00001500
      GAML=GAM(KMAX)                                                                 00001510
      FLS(4)=0.                                                                      00001520
      CALL TLOAD(KMAX)                                                               00001530
      DO 14 M=1,MNMAX                                                                00001540
      IF(M.GT.1) ELLS(1)=0.0                                                         00001550
      IF(ITRMAX.EQ.1) GO TO 68                                                       00001560
      FLS(1)=-TT(M)*ALOAD+OSE*(BX3(M)+BE3(M)+NU*(BT3(M)+BE3(M)))*BL                  00001570
      FLS(2)  =OSE*(BL*D1*BXT3(M)+EX3(M)+ET3(M))                                     00001580
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD-(EXX3(M)+ETX3(M))*SOE                          00001590
      GO TO 69                                                                       00001600
   68 FLS(1)=-TT(M)*ALOAD                                                            00001610
      FLS(2)=0.                                                                      00001620
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD                                                00001630
   69 CONTINUE                                                                       00001640
      IK=KL+KMAX*(M-1)                                                               00001650
      IJ=KMAX*M                                                                      00001660
      L=M*KMAX2                                                                      00001670
      DO 14 I=1,4                                                                    00001680
      SUMZ=0.                                                                        00001690
      DO 15 J=1,4                                                                    00001700
C*****************************************************************                 00001710
C     THE FOLLOWING CARD CAUSES BOUNDARY CONS TO EXIST FOR MODE '0' ONLY           00001720
C*****************************************************************                 00001730
      IF (M.NE.1) ELLS(J)=0                                                          00001740
   15 SUMZ=SUMZ+ZF1M(I,J,M)*ELLS(J)+ZF2M(I,J,M)*X(J,IJ)+ZF3M(I,J,M)*                00001750
     1X(J,IK)+ZF4M(I,J,M)*FLS(J)                                                   00001760
   14 Z(I,L)=SUMZ                                                                   00001770
      LS=1                                                                           00001780
  150 DO 16 M=1,MNMAX                                                               00001790
      DO 16 L=LS,KMAX                                                                00001800
```

```
      K=KMAX2-L                                                          00001810
      KPX=K-1                                                            00001820
      KZ=K+1                                                             00001830
      IJ=KPX+(M-1)*KMAX                                                  00001840
      JK=KZ+(M-1)*KMAX2                                                  00001850
      KK=JK-1                                                            00001860
      DO 17 I=1,4                                                        00001870
      SUMZ=0.                                                            00001880
      DO 18 J=1,4                                                        00001890
   18 SUMZ=SUMZ-P(I,J,IJ)*Z(J,JK)                                        00001900
      SUMZ=SUMZ+X(I,IJ)                                                  00001910
      ASUMZ=ABS(SUMZ)                                                    00001920
      IF(ASUMZ.GT.1.E+15) ITR=ITRMAX                                     00001930
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 17                     00001940
      DELZ=ABS(Z(I,KK)-SUMZ)                                             00001950
      ZTEST=EPS*TZMAX(I,M)                                               00001960
      IF(DELZ.GT.ZTEST) NCONV=0                                          00001970
   17 Z(I,KK)=SUMZ                                                       00001980
   16 CONTINUE                                                           00001990
      IF(IBCINL.LT.0) GO TO 30                                           00002000
      DO 25 M=1,MNMAX                                                    00002010
      CALL EFG(1,M)                                                      00002020
      CALL ABC                                                           00002030
      IJ=2+(M-1)*KMAX2                                                   00002040
      IJ1=IJ+1                                                           00002050
      IJ2=IJ-1                                                           00002060
      DO 21 I=1,4                                                        00002070
      SUMZ=0.                                                            00002080
      DO 22 J=1,4                                                        00002090
   22 SUMZ=SUMZ-A(I,J)*Z(J,IJ1)-BEE(I,J)*Z(J,IJ)                         00002100
   21 ZT(I)=SUMZ+GEES(I,M)                                               00002110
      CALL MATINV(C,4,ZT,1,DETERM,IPIVOT,INDEX,4,ISCALE)                 00002120
      DO 23 I=1,4                                                        00002130
   23 Z(I,IJ2)=ZT(I)                                                     00002140
   25 CONTINUE                                                           00002150
   30 RETURN                                                             00002160
  100 CALL INLPOL                                                        00002170
      DO 101 M=1,MNMAXO                                                  00002180
      U1(M)=U2(M)                                                        00002190
      V1(M)=V2(M)                                                        00002200
      W1(M)=W2(M)                                                        00002210
      IJ=3+KMAX2*(M-1)                                                   00002220
      U2(M)=Z(1,IJ)                                                      00002230
      V2(M)=Z(2,IJ)                                                      00002240
  101 W2(M)=Z(3,IJ)                                                      00002250
      GO TO 102                                                          00002260
  120 IF(ITRMAX.NE.1) CALL FNLPOL                                        00002270
      CALL FORCE(KL)                                                     00002280
```

```
      IF(M2.EQ.0) GO TO 122                                                    00002290
      L=KL+(M2-1)*KMAX                                                         00002300
      L1=KMAX1+(M2-1)*KMAX2                                                    00002310
      DO 130 I=1,4                                                             00002320
      SUM=0.                                                                   00002330
      DO 131 J=1,4                                                             00002340
  131 SUM=SUM+CL2(I,J)*X(J,L)                                                  00002350
      ASUMZ=ABS(SUM)                                                           00002360
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TO 130                            00002370
      DELZ=ABS(Z(I,L1)-SUM)                                                    00002380
      ZTEST=EPS*TZMAX(I,M2)                                                    00002390
      IF(DELZ.GT.ZTEST) NCONV=0                                                00002400
  130 Z(I,L1)=SUM                                                              00002410
  122 IF(M1.EQ.0) GO TO 123                                                    00002420
      L=KL+(M1-1)*KMAX                                                         00002430
      L1=KMAX1+(M1-1)*KMAX2                                                    00002440
      DO 132 I=1,4                                                             00002450
      SUM=0.                                                                   00002460
      DO 133 J=1,4                                                             00002470
  133 SUM=SUM+CL1(I,J)*X(J,L)                                                  00002480
      ASUMZ=ABS(SUM)                                                           00002490
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 132                          00002500
      DELZ=ABS(Z(I,L1)-SUM)                                                    00002510
      ZTEST=EPS*TZMAX(I,M1)                                                    00002520
      IF(DELZ.GT.ZTEST) NCONV=0                                                00002530
  132 Z(I,L1)=SUM                                                              00002540
  123 IF(M0.EQ.0) GO TO 124                                                    00002550
      L=KL+(M0-1)*KMAX                                                         00002560
      L1=KMAX1+(M0-1)*KMAX2                                                    00002570
      DO 134 I=1,4                                                             00002580
      SUM=0.                                                                   00002590
      DO 135 J=1,4                                                             00002600
  135 SUM=SUM+CL0(I,J)*X(J,L)                                                  00002610
      ASUMZ=ABS(SUM)                                                           00002620
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-06) GO TO 134                            00002630
      DELZ=ABS(Z(I,L1)-SUM)                                                    00002640
      ZTEST=EPS*TZMAX(I,M0)                                                    00002650
      IF(DELZ.GT.ZTEST) NCONV=0                                                00002660
  134 Z(I,L1)=SUM                                                              00002670
  124 LS=2                                                                     00002680
      GO TO 150                                                                00002690
      END                                                                      00002700
```

```
      SUBROUTINE PLOT1(I)                                                    00000010
C*************************************************************************00000020
C     THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER    *00000030
C     SPECIFIED) INPUT QUANTITIES                                      *00000040
C*************************************************************************00000050
      IMPLICIT LOGICAL*1 ($)                                                 00000060
      COMMON /IBL4/ KMAX,KL                                                  00000070
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,      00000080
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,      00000090
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,         00000100
     3                IPHIT,IPHI,$PLOTS,$MODAL                              00000110
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),       00000120
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),       00000130
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),       00000140
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),       00000150
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),     00000160
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),         00000170
     6                YPHI(100),XSTATN(100)                                  00000180
C*************************************************************************00000190
      NGKMAX=-KMAX                                                           00000200
      IF (I.GT.1) GO TO 1                                                    00000210
      IF (IRADII.EQ.0) GO TO 2                                               00000220
      WRITE (6,1000)                                                         00000230
      CALL PLOTIT (XSTATN,XRADII,NGKMAX,0)                                   00000240
      WRITE (6,1001)                                                         00000250
    2 IF (IGAMMA.EQ.0) GO TO 4                                               00000260
      WRITE (6,1000)                                                         00000270
      CALL PLOTIT (XSTATN,YGAMMA,NGKMAX,0)                                   00000280
      WRITE (6,1002)                                                         00000290
    4 IF (IOMEGS.EQ.0) GO TO 5                                               00000300
      WRITE (6,1000)                                                         00000310
      CALL PLOTIT (XSTATN,YOMEGS,NGKMAX,0)                                   00000320
      WRITE (6,1003)                                                         00000330
    5 IF (IOMEGT.EQ.0) GO TO 6                                               00000340
      WRITE (6,1000)                                                         00000350
      CALL PLOTIT (XSTATN,YOMEGT,NGKMAX,0)                                   00000360
      WRITE (6,1004)                                                         00000370
    6 IF (IDEOMS.EQ.0) GO TO 7                                               00000380
      WRITE (6,1000)                                                         00000390
      CALL PLOTIT (XSTATN,YDEOMS,NGKMAX,0)                                   00000400
      WRITE (6,1005)                                                         00000410
    7 IF (IBSTIF.EQ.0) GO TO 8                                               00000420
```

```
      WRITE (6,1000)                                                    00000430
      CALL PLOTIT (XSTATN,YBSTIF,NGKMAX,0)                              00000440
      WRITE (6,1006)                                                    00000450
    8 IF (IDSTIF.EQ.0) GO TO 9                                          00000460
      WRITE (6,1000)                                                    00000470
      CALL PLOTIT (XSTATN,YDSTIF,NGKMAX,0)                              00000480
      WRITE (6,1007)                                                    00000490
    9 IF (IBBSTF.EQ.0) GO TO 10                                         00000500
      WRITE (6,1000)                                                    00000510
      CALL PLOTIT (XSTATN,YBBSTF,NGKMAX,0)                              00000520
      WRITE (6,1008)                                                    00000530
   10 IF (IDDSTF.EQ.0) GO TO 1                                          00000540
      WRITE (6,1000)                                                    00000550
      CALL PLOTIT (XSTATN,YDDSTF,NGKMAX,0)                              00000560
      WRITE (6,1009)                                                    00000570
    1 IF (IPR.EQ.0) GO TO 11                                            00000580
      WRITE (6,1000)                                                    00000590
      CALL PLOTIT (XSTATN,YPR,NGKMAX,0)                                 00000600
      WRITE (6,1010)                                                    00000610
   11 IF (IPS.EQ.0) GO TO 12                                            00000620
      WRITE (6,1000)                                                    00000630
      CALL PLOTIT (XSTATN,YPS,NGKMAX,0)                                 00000640
      WRITE (6,1011)                                                    00000650
   12 IF (IPT.EQ.0) GO TO 13                                            00000660
      WRITE (6,1000)                                                    00000670
      CALL PLOTIT (XSTATN,YPT,NGKMAX,0)                                 00000680
      WRITE (6,1012)                                                    00000690
   13 IF (ITT.EQ.0) GO TO 14                                            00000700
      WRITE (6,1000)                                                    00000710
      CALL PLOTIT (XSTATN,YTT,NGKMAX,0)                                 00000720
      WRITE (6,1013)                                                    00000730
   14 IF (IMT.EQ.0) GO TO 15                                            00000740
      WRITE (6,1000)                                                    00000750
      CALL PLOTIT (XSTATN,YMT,NGKMAX,0)                                 00000760
      WRITE (6,1014)                                                    00000770
   15 IF (IDTT.EQ.0) GO TO 16                                           00000780
      WRITE (6,1000)                                                    00000790
      CALL PLOTIT (XSTATN,YDTT,NGKMAX,0)                                00000800
      WRITE (6,1015)                                                    00000810
   16 IF (IDMT.EQ.0) GO TO 17                                           00000820
      WRITE (6,1000)                                                    00000830
      CALL PLOTIT (XSTATN,YDMT,NGKMAX,0)                                00000840
      WRITE (6,1016)                                                    00000850
      WRITE (6,1000)                                                    00000860
   17 RETURN                                                            00000870
C*****************************************************************     00000880
 1000 FORMAT ('1')                                                      00000890
 1001 FORMAT ('0',T10,'RADIUS VS STATION')                              00000900
```

```
 1002 FORMAT ('0',T10,'GAMMA VS STATION')                                  00000910
 1003 FORMAT ('0',T10,'OMEGA-S VS STATION')                                00000920
 1004 FORMAT ('0',T10,'OMEGA-THETA VS STATION')                            00000930
 1005 FORMAT ('0',T10,'DEOMEGA-S VS STATION')                              00000940
 1006 FORMAT ('0',T10,'B-STIFFNESS VS STATION')                            00000950
 1007 FORMAT ('0',T10,'D-STIFFNESS VS STATION')                            00000960
 1008 FORMAT ('0',T10,'DB-STIFFNESS VS STATION')                           00000970
 1009 FORMAT ('0',T10,'DD-STIFFNESS VS STATION')                           00000980
 1010 FORMAT ('0',T10,'NORMAL LOADING VS STATION')                         00000990
 1011 FORMAT ('0',T10,'MERIDIO-TANGENTIAL LOADING VS STATION')             00001000
 1012 FORMAT ('0',T10,'CIRCUMFERENTIO-TANGENTIAL LOADING VS STATION')      00001010
 1013 FORMAT ('0',T10,'THERMAL LOADING VS STATION')                        00001020
 1014 FORMAT ('0',T10,'THERMAL BENDING LOADING VS STATION')                00001030
 1015 FORMAT ('0',T10,'DE-THERMAL LOADING VS STATION')                     00001040
 1016 FORMAT ('0',T10,'DE-THERMAL-BENDING LOADING VS STATION')             00001050
C*************************************************************************00001060
      END                                                                  00001070
```



```
      SUBROUTINE PMATRX                                                    00000010
C*************************************************************************00000020
C         THIS SUBROUTINE CALLS THE SUBROUTINES HJ(K,MN), EFG(K,MN), ABC,00000030
C         AND PANDD(K,MN) TO SET UP THE P, P-BAR AND P-HAT MATRICES GIVEN00000040
C         BY EQUATIONS (30) OF REFERENCE (3).                           *00000050
C         INTERNALLY, MATRICES DL, DG AND DF ARE SET UP FOR THE CALCULA-*00000060
C         TION OF X(1) GIVEN BY EQUATION (31A) OF REF. (3), WHERE       *00000070
C                                                                       *00000080
C         X(1) = DL*SMALL-L(1) + DG*SMALL-G(1) + DF*SMALL-F(1)          *00000090
C                                                                       *00000100
C         THE SPECIAL P MATRIX FOR A SHELL WITH AN INITIAL POLE IS ALSO *00000110
C         COMPUTED HERE.                                                *00000120
C         MATRICES ZF1M,ZF2M,ZF3M,ZF4M ARE SET UP FOR THE CALCULATION OF*00000130
C         Z(K+1) GIVEN BY EQUATION (31B) OF REF. (3), WHERE             *00000140
C                                                                       *00000150
C         Z(K+1)=ZF1M*SMALL-L(K) + ZF2M*X(K) + ZF3M*X(K-1) + ZF4M*SMALL-F00000160
C                                                                       *00000170
C         IF THE SHELL HAS A FINAL POLE, THE MATRICES CL0,CL1,CL2 ARE   *00000180
C         PREPARED FOR THE CALCULATION OF Z(K)                          *00000190
```

```
C*******************************************************************00000200
      REAL JAY                                                          00000210
      COMMON /IBL1/ MNMAX                                               00000220
      COMMON /IBL2/ N(10),MNINIT                                        00000230
      COMMON /IBL3/ M0,M1,M2,M3                                         00000240
      COMMON /IBL4/ KMAX,KL                                             00000250
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000260
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              00000270
      COMMON /BL4/  P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),      00000280
     1              ZF3M(4,4,10),ZF4M(4,4,10)                           00000290
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),        00000300
     1              UNIT(4,4)                                           00000310
      COMMON /BL23/ JAY(4,4),H(4,4)                                     00000320
      COMMON /BL24/ DL(4,4,10),DG(4,4,10),DF(4,4,10)                    00000330
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                00000340
      DIMENSION PATA(4,4),PBTA(4,4),POTA(4,4),PJTA(4,4),DLL(4,4),PTR(4, 00000350
     14),DGG(4,4),ZF1(4,4),ZF2(4,4),ZFP0(4,4),ZFP1(4,4),ZFP2(4,4),IPIVO 00000360
     2T(4),INDEX(4,2),CL0(4,4),CL1(4,4),CL2(4,4),G1(4)                  00000370
      EQUIVALENCE (CL0(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1)),   00000380
     1(ZFP0(1),PATA(1)),(ZFP1(1),PBTA(1)),(ZFP2(1),POTA(1))             00000390
     2,(ZF1(1),DLL(1)),(ZF2(1),PTR(1))                                  00000400
C*******************************************************************00000410
      IF(IBCINL.LT.0) GO TO 10                                          00000420
      DO 1 MN=MNINIT,MNMAX                                              00000430
      CALL HJ(1,MN)                                                     00000440
      CALL EFG(1,MN)                                                    00000450
      CALL ABC                                                          00000460
      CALL MATINV(C,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)                00000470
      DO 3 I=1,4                                                        00000480
      DO 3 J=1,4                                                        00000490
      SUMA=0.                                                           00000500
      SUMB=0.                                                           00000510
      SUMO=0.                                                           00000520
      SUMJ=0.                                                           00000530
      DO 4 L=1,4                                                        00000540
      SUMJ=SUMJ+OMEG1(I,L)*JAY(L,J)                                     00000550
      SUMA=SUMA+C(I,L)*A(L,J)                                           00000560
      SUMB=SUMB+C(I,L)*BEE(L,J)                                         00000570
    4 SUMO=SUMO+OMEG1(I,L)*H(L,J)                                       00000580
      PATA(I,J)=SUMA+UNIT(I,J)                                          00000590
      PBTA(I,J)=SUMB                                                    00000600
      POTA(I,J)=SUMO                                                    00000610
    3 PJTA(I,J)=SUMJ                                                    00000620
      DO 5 I=1,4                                                        00000630
      DO 5 J=1,4                                                        00000640
      SUMOB=0.                                                          00000650
      SUMOA=0.                                                          00000660
      SUMOC=0.                                                          00000670
```

```
      DO 6 L=1,4                                                        00000680
      SUMOB=SUMOB+POTA(I,L)*PBTA(L,J)                                   00000690
      SUMOA=SUMOA+POTA(I,L)*PATA(L,J)                                   00000700
    6 SUMOC=SUMOC+POTA(I,L)*C(L,J)                                      00000710
      DLL(I,J)=SUMOB+PJTA(I,J)+CAPL1(I,J)                               00000720
      PTR(I,J)=SUMOA                                                    00000730
    5 DGG(I,J)=SUMOC                                                    00000740
      CALL MATINV(DLL,4,PTR,4,DETERM,IPIVOT,INDEX,4,ISCALE)             00000750
      DO 1 I=1,4                                                        00000760
      DO 1 J=1,4                                                        00000770
      SUMD=0.                                                           00000780
      SUME=0.                                                           00000790
      DO 7 L=1,4                                                        00000800
      SUMD=SUMD+DLL(I,L)*DGG(L,J)                                       00000810
    7 SUME=SUME-DLL(I,L)*OMEG1(L,J)                                     00000820
      DL(I,J,MN)=DLL(I,J)                                               00000830
      DG(I,J,MN)=SUMD                                                   00000840
      DF(I,J,MN)=SUME                                                   00000850
      IJ=1+KMAX*(MN-1)                                                  00000860
    1 P(I,J,IJ)=PTR(I,J)                                                00000870
      GO TO 20                                                          00000880
   10 DO 11 MN=MNINIT,MNMAX                                             00000890
      NN=IABS(N(MN))                                                    00000900
      IJ=1+KMAX*(MN-1)                                                  00000910
      DO 14 I=1,4                                                       00000920
      X(I,IJ)=0.                                                        00000930
      DO 14 J=1,4                                                       00000940
   14 P(I,J,IJ)=0.                                                      00000950
      IF(NN.GT.3) GO TO 11                                              00000960
      IF(NN.GT.2) GO TO 90                                              00000970
      IF(NN.GT.1) GO TO 12                                              00000980
      IF(NN.GT.0) GO TO 13                                              00000990
      P(3,3,IJ)=-1.                                                     00001000
      P(4,4,IJ)=-1.                                                     00001010
      M0=MN                                                             00001020
      GO TO 11                                                          00001030
   12 P(4,4,IJ)=-1.                                                     00001040
      M2=MN                                                             00001050
      GO TO 11                                                          00001060
   90 M3=MN                                                             00001070
      GO TO 11                                                          00001080
   13 P(1,1,IJ)=-1.                                                     00001090
      P(2,1,IJ)=1.                                                      00001100
      IF(N(MN).LT.0) P(2,1,IJ)=-1.                                      00001105
      M1=MN                                                             00001110
   11 CONTINUE                                                          00001120
   20 KLAST=KMAX                                                        00001130
      IF(IBCFNL.LT.0) KLAST=KL                                          00001140
```

```
      DO 23 K=2,KLAST                                                    00001150
      DO 23 MN=MNINIT,MNMAX                                              00001160
      CALL EFG(K,MN)                                                     00001170
      CALL ABC                                                           00001180
   23 CALL PANDD(K,MN)                                                   00001190
      IF(IBCFNL.LT.0) GO TO 30                                           00001200
      DO 40 MN=MNINIT,MNMAX                                              00001210
      IKL=MN*KMAX-1                                                      00001220
      JKL=KMAX*MN                                                        00001230
      CALL HJ(KMAX,MN)                                                   00001240
      DO 41 I=1,4                                                        00001250
      DO 41 J=1,4                                                        00001260
      SUMO=0.                                                            00001270
      SUMP=0.                                                            00001280
      SUMJ=0.                                                            00001290
      DO 42 L=1,4                                                        00001300
      SUMO=SUMO+OMEGL(I,L)*H(L,J)                                        00001310
      SUMP=SUMP+P(I,L,IKL)*P(L,J,JKL)                                    00001320
   42 SUMJ=SUMJ+OMEGL(I,L)*JAY(L,J)                                      00001330
      PATA(I,J)=SUMO                                                     00001340
      PBTA(I,J)=UNIT(I,J)-SUMP                                           00001350
   41 PJTA(I,J)=SUMJ+CAPLL(I,J)                                          00001360
      DO 43 I=1,4                                                        00001370
      DO 43 J=1,4                                                        00001380
      SUMOP=0.                                                           00001390
      SUMJP=0.                                                           00001400
      SUMOM=0.                                                           00001410
      DO 44 L=1,4                                                        00001420
      SUMOP=SUMOP+PATA(I,L)*PBTA(L,J)                                    00001430
      SUMJP=SUMJP+PJTA(I,L)*P(L,J,JKL)                                   00001440
   44 SUMOM=SUMOM-PATA(I,L)*P(L,J,IKL)                                   00001450
      ZF1(I,J)=SUMOP-SUMJP                                               00001460
   43 ZF2(I,J)=SUMOM-PJTA(I,J)                                           00001470
      CALL MATINV(ZF1,4,ZF2,4,DETERM,IPIVOT,INDEX,4,ISCALE)              00001480
      DO 45 I=1,4                                                        00001490
      DO 45 J=1,4                                                        00001500
      SZF3=0.                                                            00001510
      SZF4=0.                                                            00001520
      DO 46 L=1,4                                                        00001530
      SZF3=SZF3+ZF1(I,L)*PATA(L,J)                                       00001540
   46 SZF4=SZF4-ZF1(I,L)*OMEGL(L,J)                                      00001550
      ZF3M(I,J,MN)=SZF3                                                  00001560
      ZF4M(I,J,MN)=SZF4                                                  00001570
      ZF1M(I,J,MN)=ZF1(I,J)                                              00001580
   45 ZF2M(I,J,MN)=ZF2(I,J)                                              00001590
   40 CONTINUE                                                           00001600
      RETURN                                                             00001610
   30 DO 31 MN=MNINIT,MNMAX                                              00001620
```

```
      IKL=MN*KMAX-1                                                     00001630
      NN=IABS(N(MN))                                                    00001640
      IF(NN    .GT.3) GO TO 31                                          00001650
      IF(NN.GT.2) GO TO 300                                             00001660
      IF(NN    .GT.1) GO TO 33                                          00001670
      IF(NN    .GT.0) GO TO 34                                          00001680
      MO=MN                                                             00001690
      DO 35 J=1,4                                                       00001700
      DO 35 I=1,4                                                       00001710
      CLO(I,J)=0.                                                       00001720
   35 ZFPO(I,J)=0.                                                      00001730
      ZFPO(1,1)=1.                                                      00001740
      ZFPO(2,2)=1.                                                      00001750
      ZFPO(3,1)=P(3,1,IKL)                                              00001760
      ZFPO(3,2)=P(3,2,IKL)                                              00001770
      ZFPO(3,3)=P(3,3,IKL)+1.                                           00001780
      ZFPO(3,4)=P(3,4,IKL)                                              00001790
      ZFPO(4,1)=P(4,1,IKL)                                              00001800
      ZFPO(4,2)=P(4,2,IKL)                                              00001810
      ZFPO(4,3)=P(4,3,IKL)                                              00001820
      ZFPO(4,4)=P(4,4,IKL)+1.                                           00001830
      CLO(3,3)=1.                                                       00001840
      CLO(4,4)=1.                                                       00001850
      CALL MATINV(ZFPO,4,CLO,4,DETERM,IPIVOT,INDEX,4,ISCALE)            00001860
      GO TO 31                                                          00001870
  300 M3=MN                                                             00001880
      GO TO 31                                                          00001890
   34 M1=MN                                                             00001900
      DO 60 J=1,4                                                       00001910
      DO 60 I=1,4                                                       00001920
      CL1(I,J)=0.                                                       00001930
   60 ZFP1(I,J)=0.                                                      00001940
      ZFP1(1,1)=P(1,1,IKL)+1.                                           00001950
      ZFP1(1,2)=P(1,2,IKL)                                              00001960
      ZFP1(1,3)=P(1,3,IKL)                                              00001970
      ZFP1(1,4)=P(1,4,IKL)                                              00001980
      ZFP1(2,1)=1.                                                      00001990
      ZFP1(2,2)=-1.                                                     00002000
      IF(N(MN).LT.0) ZFP1(2,2)=1.                                       00002005
      ZFP1(3,3)=1.                                                      00002010
      ZFP1(4,4)=1.                                                      00002020
      CL1(1,1)=1.                                                       00002030
      CALL MATINV(ZFP1,4,CL1,4,DETERM,IPIVOT,INDEX,4,ISCALE)            00002040
      GO TO 31                                                          00002050
   33 M2=MN                                                             00002060
      DO 70 J=1,4                                                       00002070
      DO 70 I=1,4                                                       00002080
      CL2(I,J)=0.                                                       00002090
```

```
   70 ZFP2(I,J)=0.                                                          00002100
      ZFP2(1,1)=1.                                                          00002110
      ZFP2(2,2)=1.                                                          00002120
      ZFP2(3,3)=1.                                                          00002130
      ZFP2(4,1)=P(4,1,IKL)                                                  00002140
      ZFP2(4,2)=P(4,2,IKL)                                                  00002150
      ZFP2(4,3)=P(4,3,IKL)                                                  00002160
      ZFP2(4,4)=P(4,4,IKL)+1.                                               00002170
      CL2(4,4)=1.                                                           00002180
      CALL MATINV(ZFP2,4,CL2,4,DETERM,IPIVOT,INDEX,4,ISCALE)                00002190
   31 CONTINUE                                                              00002200
      RETURN                                                                00002210
      END                                                                   00002220
```



```
      SUBROUTINE UPDATE                                                     00000010
C*************************************************************************00000020
C        THIS SUBROUTINE UPDATES THE STORAGE LOCATIONS OF THE BETA'S AND  00000030
C        ETA'S.  IT IS CALLED IN SUBROUTINE XANDZ AFTER A MERIDIAN       *00000040
C        STATION CHANGE.                                                 *00000050
C*************************************************************************00000060
      COMMON /IBL1/ MNMAX                                                   00000070
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                        00000080
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)     00000090
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),       00000100
     1              BXT2(10),BE2(10)                                        00000110
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),    00000120
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)              00000130
      COMMON /BL31/ DELSQ,EXT1(10)                                          00000140
C*************************************************************************00000150
      DO 1 M=1,MNMAX                                                        00000160
      BX1(M)=BX2(M)                                                         00000170
      BT1(M)=BT2(M)                                                         00000180
      BXT1(M)=BXT2(M)                                                       00000190
      BE1(M)=BE2(M)                                                         00000200
      BX2(M)=BX3(M)                                                         00000210
      BT2(M)=BT3(M)                                                         00000220
      BXT2(M)=BXT3(M)                                                       00000230
```

```
      BE2(M)=BE3(M)                                                     00000240
      EXX1(M)=EXX2(M)                                                   00000250
      ETT1(M)=ETT2(M)                                                   00000260
      ETX1(M)=ETX2(M)                                                   00000270
      EXT1(M)=EXT2(M)                                                   00000280
      EX1(M)=EX2(M)                                                     00000290
      ET1(M)=ET2(M)                                                     00000300
      EXX2(M)=EXX3(M)                                                   00000310
      ETT2(M)=ETT3(M)                                                   00000320
      ETX2(M)=ETX3(M)                                                   00000330
      EXT2(M)=EXT3(M)                                                   00000340
      EX2(M)=EX3(M)                                                     00000350
    1 ET2(M)=ET3(M)                                                     00000360
      RETURN                                                            00000370
      END                                                               00000380
```



```
      SUBROUTINE INLPOL                                                 00000010
C*****************************************************************    00000020
C         THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *00000030
C         -SUB THETA, -SUB S-THETA, AETA-SUB S-S AND -SUB THETA-S AT AN *00000040
C         INITIAL POLE.                                                 *00000050
C*****************************************************************    00000060
      COMMON /IBL1/ MNMAX                                               00000070
      COMMON /IBL3/ M0,M1,M2,M3                                         00000080
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000090
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000100
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)                       00000110
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                              00000120
      COMMON /BL7/  D1,S1                                               00000130
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                00000140
      COMMON /BL17/ DEL                                                 00000150
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),   00000160
     1              BXT2(10),BE2(10)                                    00000170
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),00000180
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)          00000190
      COMMON /BL31/ DELSQ,EXT1(10)                                      00000200
C*****************************************************************    00000210
```

```
      DO 1 MN=1,MNMAX                                                    00000220
      BX1 (MN)=0.                                                        00000230
      BT1 (MN)=0.                                                        00000240
      BXT1(MN)=0.                                                        00000250
      BE1 (MN)=0.                                                        00000260
      EX1 (MN)=0.                                                        00000270
      ET1 (MN)=0.                                                        00000280
      ETX1(MN)=0.                                                        00000290
    1 EXX1(MN)=0.                                                        00000300
      IF(M1.EQ.0) RETURN                                                 00000310
      I2=2+(M1-1)*KMAX2                                                  00000320
      I3=I2+1                                                            00000330
      I4=I3+1                                                            00000340
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z(1,I2)       00000350
      BET=.5*PHEE**2                                                     00000360
      IF(ITRMAX.EQ.1) BET=0.                                             00000370
      T2=0.                                                              00000380
      IF(M2.EQ.0) GO TO 2                                                00000390
      CALL BDB(1,B,DB,D,DD)                                              00000400
      I2=2+(M2-1)*KMAX2                                                  00000410
      I3=I2+1                                                            00000420
      I4=I3+1                                                            00000430
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BET)      00000440
      Q1=.5*PHEE*T2                                                      00000450
      BX1(M2)=BET                                                        00000460
      BT1(M2)=-BET                                                       00000470
      BXT1(M2)=-BET                                                      00000480
      ETX1(M1)=Q1                                                        00000490
      IF(M3.EQ.0) GO TO 2                                                00000500
      EXX1(M3)=Q1                                                        00000510
      ETX1(M3)=-Q1                                                       00000520
    2 T0=0.                                                              00000530
      IF(M0.EQ.0) GO TO 3                                                00000540
      BX1(M0)=BET                                                        00000550
      BT1(M0)=BET                                                        00000560
      CALL BDB(1,B,DB,D,DD)                                              00000570
      CALL TLOAD(1)                                                      00000580
      I2=2+(M0-1)*KMAX2                                                  00000590
      I3=I2+1                                                            00000600
      I4=I3+1                                                            00000610
      T0=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I2)  00000620
     1+.5*SOE*BET)-TT(M0)*ALOAD                                          00000630
    3 EXX1(M1)=PHEE*(T0+.5*T2)                                           00000640
      RETURN                                                             00000650
      END                                                                00000660
```

```
      SUBROUTINE FNLPOL                                                 00000010
C*****************************************************************    00000020
C         THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *00000030
C         -SUB THETA, -SUB S-THETA, AETA-SUB S-S, AND -SUB THETA-S AT A *00000040
C         FINAL POLE.                                                   *00000050
C*****************************************************************    00000060
      COMMON /IBL1/ MNMAX                                               00000070
      COMMON /IBL3/ M0,M1,M2,M3                                         00000080
      COMMON /IBL4/ KMAX,KL                                             00000090
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000100
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000110
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)                       00000120
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                              00000130
      COMMON /BL7/  D1,S1                                               00000140
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                00000150
      COMMON /BL17/ DEL                                                 00000160
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    00000170
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 00000180
C*****************************************************************    00000190
      DO 1 MN=1,MNMAX                                                   00000200
      BX3 (MN)=0.                                                       00000210
      BT3 (MN)=0.                                                       00000220
      BXT3(MN)=0.                                                       00000230
      BE3 (MN)=0.                                                       00000240
      EX3 (MN)=0.                                                       00000250
      ET3 (MN)=0.                                                       00000260
      ETX3(MN)=0.                                                       00000270
    1 EXX3(MN)=0.                                                       00000280
      CALL BDB(KMAX,B,DB,D,DB)                                          00000290
      IF(M1.EQ.0) RETURN                                                00000300
      KM=KMAX1+(M1-1)*KMAX2                                             00000310
      KM1=KM-1                                                          00000320
      KM2=KM-2                                                          00000330
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(KMAX)*Z(1,KM)00000340
      BET=.5*PHEE**2                                                    00000350
      IF(ITRMAX.EQ.1) BET=0.                                            00000360
      T2=0.                                                             00000370
      IF(M2.EQ.0) GO TO 2                                               00000380
      KM=KMAX1+(M2-1)*KMAX2                                             00000390
```

```
      KM1=KM-1                                                          00000400
      KM2=KM-2                                                          00000410
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SOE*BET)    00000420
      Q1=.5*PHEE*T2                                                     00000430
      BX3(M2)=BET                                                       00000440
      BT3(M2)=-BET                                                      00000450
      BXT3(M2)= BET                                                     00000460
      ETX3(M1)=Q1                                                       00000470
      IF(M3.EQ.0) GO TO 2                                               00000480
      EXX3(M3)=Q1                                                       00000490
      ETX3(M3)=-Q1                                                      00000500
    2 T0=0.                                                             00000510
      IF(M0.EQ.0) GO TO 3                                               00000520
      CALL TLOAD(KMAX)                                                  00000530
      KM=KMAX1+(M0-1)*KMAX2                                             00000540
      KM1=KM-1                                                          00000550
      KM2=KM-2                                                          00000560
      BX3(M0)=BET                                                       00000570
      BT3(M0)=BET                                                       00000580
      T0=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+OMXI(KMAX)*    00000590
     1Z(3,KM)+.5*SOE*BET)-TT(M0)*ALOAD                                  00000600
    3 EXX3(M1)=PHEE*(T0+.5*T2)                                          00000610
      RETURN                                                            00000620
      END                                                               00000630
```



```
      SUBROUTINE PHIBET(K)                                              00000010
C*****************************************************************    00000020
C       THIS SUBROUTINE CALCULATES THE PHI'S AND CARRIES OUT THE      *00000030
C       MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE BETA    *00000040
C       NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*00000050
C       IS, JS, ID, JS,IJS, MAXS, MAXD, MAXSY ARE PREPARED IN SUB-    *00000060
C       ROUTINE MODES AND USED HERE.                                  *00000070
C*****************************************************************    00000080
      COMMON /IBL1/ MNMAX                                               00000090
      COMMON /IBL2/ N(10),MNINIT                                        00000100
      COMMON /IBL4/ KMAX,KL                                             00000110
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       00000120
```

```
     1               JS(10,10),ID(10,10),JD(10,10),IJS(10)                    00000130
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                        00000140
      COMMON /IBL13/ ITRMAX,LSMAX                                             00000150
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                                    00000160
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                  00000170
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                                 00000180
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                      00000190
      COMMON /BL12/ TDLI,TDEL                                                 00000200
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),      00000210
     1              V3(10),W3(10)                                             00000220
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                          00000230
C*****************************************************************          00000240
      OX=OMXI(K)                                                              00000250
      OT=OMT(K)                                                               00000260
      RRA=1./R(K)                                                             00000270
      GA=GAM(K)                                                               00000280
      KP2=K+2                                                                 00000290
      DO 1 M=1,MNMAXO                                                         00000300
      EN=N(M)                                                                 00000310
      IK=KP2+(M-1)*KMAX2                                                      00000320
      U3(M)=Z(1,IK)                                                           00000330
      V3(M)=Z(2,IK)                                                           00000340
      W3(M)=Z(3,IK)                                                           00000350
      PHIX(M)=-TDLI*(W3(M)-W1(M))+OX*U2(M)                                    00000360
      PHIT(M)=EN*W2(M)*RRA+V2(M)*OT                                           00000370
    1 PHI(M)=(TDLI*(V3(M)-V1(M))+GA*V2(M)+EN*U2(M)*RRA)*.5                    00000380
      IF(ITRMAX.EQ.1) RETURN                                                  00000390
      DO 9 M=1,MNMAX                                                          00000400
      SMO=0.                                                                  00000410
      SMT=0.                                                                  00000420
      SMR=0.                                                                  00000430
      SMF=0.                                                                  00000440
      IF(N(M).EQ.0) GO TO 20                                                  00000450
      MAXL=MAXS(M)                                                            00000460
      IF(MAXL.EQ.0) GO TO 2                                                   00000470
      DO 3 L=1,MAXL                                                           00000480
      I=IS(L,M)                                                               00000490
      J=JS(L,M)                                                               00000500
      SMO=SMO+PHIX(I)*PHIX(J)                                                 00000510
      SMT=SMT-PHIT(I)*PHIT(J)                                                 00000520
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                                 00000530
    3 SMF=SMF-PHI(I)*PHI(J)                                                   00000540
    2 MAXL=MAXD(M)                                                            00000550
      IF(MAXL.EQ.0) GO TO 4                                                   00000560
      DO 5 L=1,MAXL                                                           00000570
      I=ID(L,M)                                                               00000580
      J=JD(L,M)                                                               00000590
      SMO=SMO+PHIX(I)*PHIX(J)                                                 00000600
```

```
      SMT=SMT+PHIT(I)*PHIT(J)                                          00000610
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                          00000620
    5 SMF=SMF+PHI(I)*PHI(J)                                            00000630
    4 IF(MAXSY(M).EQ.0) GO TO 10                                       00000640
      I=IJS(M)                                                         00000650
      SMO=SMO+PHIX(I)**2/2.                                            00000660
      SMT=SMT-PHIT(I)**2/2.                                            00000670
      SMR=(SMR+PHIX(I)*PHIT(I))                                        00000680
      SMF=SMF-PHI(I)**2/2.                                             00000690
      GO TO 10                                                         00000700
   20 DO 21 L=1,MNMAXO                                                 00000710
      SMO=SMO+PHIX(L)**2                                               00000720
      SMT=SMT+PHIT(L)**2                                               00000730
   21 SMF=SMF+PHI(L)**2                                                00000740
      IF(M.GT.MNMAXO) GO TO 11                                         00000750
      SMO=SMO+PHIX(M)**2                                               00000760
   11 BX3(M)=SMO*.5                                                    00000770
      BT3(M)=SMT*.5                                                    00000780
      BE3(M)=SMF*.5                                                    00000790
      BXT3(M)=0.                                                       00000800
      GO TO 9                                                          00000810
   10 BX3(M)=SMO                                                       00000820
      BT3(M)=SMT                                                       00000830
      BXT3(M)=SMR*.5                                                   00000840
      BE3(M)=SMF                                                       00000850
    9 CONTINUE                                                         00000860
      RETURN                                                           00000870
      END                                                              00000880
```



```
      SUBROUTINE EFG(K,MN)                                             00000010
C*************************************************************************00000020
C         THIS SUBROUTINE PREPARES THE ELEMENTS OF THE E, F, AND G      *00000030
C         FOR EACH MERIDIAN STATION K AND FOR EACH FOURIER MODE, MN.    *00000040
C*************************************************************************00000050
      IMPLICIT LOGICAL*1 (L)                                           00000060
      REAL NU,N,    LAM2,LSD18,LSD1N,MASS,MAS                          00000070
      COMMON /IBL2/NN(10),MNINIT                                       00000080
```

```
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                    00000090
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                        00000100
      COMMON /BL14/ LAM2,LSD18,LSD1N                                            00000110
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),        00000120
     1              V3(10),W3(10)                                              00000130
      COMMON /BL20/ DEOMX(200)                                                  00000140
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                        00000150
      COMMON /BL100/ TEEO,$DYNMC                                                00000160
      COMMON /BL101/ ZO(4,220),Z2(4,220),Z3(4,220),DELSD                        00000170
      COMMON /BL102/ DELOAD                                                     00000180
      COMMON /BL103/ MASS(200)                                                  00000190
C*****************************************************************************00000200
      N=NN(MN)                                                                  00000210
      MAS=MASS(K)                                                               00000220
      CALL BDB(K,B,DB,D,DD)                                                     00000230
      E(1,1)=B                                                                  00000240
      E(1,2)=0.                                                                 00000250
      E(1,3)=0.                                                                 00000260
      E(1,4)=0.                                                                 00000270
      E(2,1)=0.                                                                 00000280
      D1=(1.-NU)                                                                00000290
      RA=R(K)                                                                   00000300
      GA=GAM(K)                                                                 00000310
      OX=OMXI(K)                                                                00000320
      OT=OMT(K)                                                                 00000330
      DEX=DEOMX(K)                                                              00000340
      REX=(3.*OT-OX)                                                            00000350
      GA2=GA**2                                                                 00000360
      RXE=(3.*OX-OT)                                                            00000370
      OTX=OT*OX                                                                 00000380
      DNLR=LAM2*D*N*D1/(2.*RA)                                                  00000390
      DDNLR=DNLR*DD/D                                                           00000400
      E(2,2)=B*D1/2.+LAM2*D*D1*REX**2/8.                                        00000410
      E(2,3)=DNLR*REX                                                           00000420
      E(2,4)=0.                                                                 00000430
      E(3,1)=0.                                                                 00000440
      E(3,2)=E(2,3)                                                             00000450
      RAN=(N/RA)**2                                                             00000460
      E(3,3)=LAM2*D*D1*(2.*RAN+(1.+NU)*GA2)                                     00000470
      E(3,4)=LAM2                                                               00000480
      E(4,1)=0.                                                                 00000490
      E(4,2)=0.                                                                 00000500
      E(4,3)=-D                                                                 00000510
      E(4,4)=0.                                                                 00000520
      F(1,1)=GA*B+DB                                                            00000530
      F(1,2)=(1.+NU)*B*N/(2.*RA)+DNLR*REX*RXE/4.                                00000540
      F(1,3)=B*(OX+NU*OT)+LAM2*D*D1*((1.+NU)*GA2*OX+RAN*RXE/2.)                 00000550
      F(1,4)=LAM2*OX                                                            00000560
```

```
 F(2,1)=-F(1,2)                                                          00000570
 F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*D*D1*REX/8.)*(2.*DEX-GA*(5.*OX           00000580
1-3.*OT))+LAM2*DD*D1*REX**2/8.                                           00000590
 F(2,3)=DNLR*(2.*(1.+NU)*GA*OT-DEX+3.*GA*(OX-OT))+DDNLR*REX              00000600
 F(2,4)=0.                                                               00000610
 F(3,1)=-F(1,3)                                                          00000620
 F(3,2)=DNLR*(3.*GA*OX-GA*OT*(5.+2.*NU)-DEX)+DDNLR*REX                   00000630
 F(3,3)=-LAM2*D*D1*((1.+NU)*(2.*GA*OX*OT+GA**3)+2.*GA*RAN)               00000640
1+LAM2*DD*D1*((1.+NU)*GA2+2.*RAN)                                        00000650
 F(3,4)=LAM2*GA*(2.-NU)                                                  00000660
 F(4,1)=D*OX                                                             00000670
 F(4,2)=0.                                                               00000680
 F(4,3)=-D*NU*GA                                                         00000690
 F(4,4)=0.                                                               00000700
 G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-D1*B*RAN/2.-LAM2*D*D1*((1.+NU)*GA2*      00000710
1OX**2+RXE**2*RAN/8.)                                                    00000720
2   - 2.*MAS/DELSD                                                       00000730
 G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-DNLR*2.*GA*(REX*RXE/8.         00000740
1+(1.+NU)*OTX)                                                           00000750
 G(1,3)=B*(DEX+GA*(OX-OT))+DB*(OX+NU*OT)-LAM2*D*D1*GA*RAN*(RXE/2.+(      00000760
11.+NU)*OX)                                                              00000770
 G(1,4)=LAM2*D1*GA*OX                                                    00000780
 G(2,1)=-B*GA*N*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.*(-1.*(1.+        00000790
1NU)*GA*OTX+GA/8.*(6.*OTX-7.*OX**2-3.*OT**2)-DEX/4.*(5.*OT-3.*OX))      00000800
2-DDNLR/4.*REX*RXE                                                       00000810
 G(2,2)=-GA*F(2,2)+D1/2.*B*OTX-B*RAN-LAM2*D*D1*((1.+NU)*OT**2*RAN        00000820
1-OTX/8.*REX**2)                                                         00000830
2   - 2.*MAS/DELSD                                                       00000840
 G(2,3)=-B*N*(OT+NU*OX)/RA+DNLR*(GA*DEX-2.*GA2*OX-2.*(1.+NU)*OT          00000850
1*RAN+REX*(GA2+OTX))-DDNLR*REX*GA                                        00000860
 G(2,4)=-NU*LAM2*OT*N/RA                                                 00000870
 G(3,1)=-B*GA*(OT+NU*OX)+LAM2*D*D1*(GA*(1.+NU)*(-GA*DEX+GA2*OX           00000880
1-OX*RAN+2.*OTX*OX)+RAN/2.*(GA*OX-GA*OT-3.*DEX))                         00000890
2-LAM2*DD*D1*((1.+NU)*GA2*OX+RAN/2.*RXE)                                 00000900
 G(3,2)=-B*N*(OT+NU*OX)/RA+DNLR*(2.*(1.+NU)*(OTX*OT-GA2*OX+2.*GA2        00000910
1*OT-OT*RAN)+GA*DEX+3.*GA2*(OT-OX)+OTX*REX)-DDNLR*(2.*(1.+NU)*GA         00000920
2*OT+GA*REX)                                                             00000930
 G(3,3)=-B*(OX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+NU)*(OTX-RAN       00000940
1+2.*GA2)+2.*(GA2+OTX))-LAM2*DD*D1*RAN*(3.+NU)*GA                        00000950
2   - 2.*MAS/DELSD                                                       00000960
 G(3,4)=-LAM2*(D1*OTX+NU*RAN)                                            00000970
 G(4,1)=D*(DEX+NU*GA*OX)                                                 00000980
 G(4,2)=D*NU*N*OT/RA                                                     00000990
 G(4,3)=D*NU*RAN                                                         00001000
 G(4,4)=-1.                                                              00001010
 RETURN                                                                  00001020
 END                                                                     00001030
```

```
      SUBROUTINE TEAETA(K)                                                  00000010
C*****************************************************************      00000020
C         THIS SUBROUTINE CALCULATES THE INPLANE FORCES AND CARRIES OUT *   00000030
C         THE MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTINE THE AETA*   00000040
C         NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*   00000050
C         IS, JS, ID, JS, IJS, MAXS, MAXD, MAXSY PREPARED IN SUBROUTINE *   00000060
C         MODES ARE USED HERE                                           *   00000070
C*****************************************************************      00000080
      REAL NU,MT                                                            00000090
      COMMON /IBL1/ MNMAX                                                   00000100
      COMMON /IBL2/ N(10),MNINIT                                            00000110
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),           00000120
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                   00000130
      COMMON /IBL8/ LSTEP,ITR                                               00000140
      COMMON /IBL13/ ITRMAX,LSMAX                                           00000150
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                            00000160
      COMMON /BL6/  Z(4,220),SOE,OSE,ALOAD                                  00000170
      COMMON /BL7/  D1,S1                                                   00000180
      COMMON /BL8/  R(200),GAM(200),OMT(200)                                00000190
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                               00000200
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                    00000210
      COMMON /BL12/ TDLI,TDEL                                               00000220
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),    00000230
     1              V3(10),W3(10)                                           00000240
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                        00000250
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)     00000260
      DIMENSION TX(10),TTH(10),TXT(10)                                      00000270
C*****************************************************************      00000280
      RRA=1./R(K)                                                           00000290
      GA=GAM(K)                                                             00000300
      OX=OMXI(K)                                                            00000310
      OT=OMT(K)                                                             00000320
      CALL BDB(K,BS,DB,DS,DD)                                               00000330
      DO 1 M=1,MNMAXO                                                       00000340
      EN=N(M)                                                               00000350
      CALL TLOAD(K)                                                         00000360
      TTS=TT(M)*ALOAD                                                       00000370
```

```
      EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+OSE*(BX3(M)+BE3(M))                    00000380
      ET= EN *V2(M)*RRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M))              00000390
      EXT=.5*(TDLI*(V3(M)-V1(M))- EN *U2(M)*RRA-GA*V2(M))+OSE*BXT3(M)      00000400
      TX(M)=BS*(EX+NU*ET)-TTS                                              00000410
      TTH(M)=BS*(ET+NU*EX)-TTS                                             00000420
    1 TXT(M)=BS*D1*EXT                                                     00000430
      DO 9 M=1,MNMAX                                                       00000440
      SMF=0.                                                               00000450
      SMS=0.                                                               00000460
      SMV=0.                                                               00000470
      SME=0.                                                               00000480
      SMN=0.                                                               00000490
      SMT=0.                                                               00000500
      IF(N(M).EQ.0) GO TO 20                                               00000510
      MAXL=MAXS(M)                                                         00000520
      IF(MAXL.EQ.0) GO TO 2                                                00000530
      DO 3 L=1,MAXL                                                        00000540
      I=IS(L,M)                                                            00000550
      J=JS(L,M)                                                            00000560
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                  00000570
      SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                                00000580
      SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)                                00000590
      SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                                00000600
      SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)                                    00000610
    3 SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)                                  00000620
    2 MAXL=MAXD(M)                                                         00000630
      IF(MAXL.EQ.0) GO TO 4                                                00000640
      DO 5 L=1,MAXL                                                        00000650
      I=ID(L,M)                                                            00000660
      J=JD(L,M)                                                            00000670
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                  00000680
      SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                                00000690
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)                                00000700
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                                00000710
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)                                    00000720
    5 SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)                                  00000730
    4 IF(MAXSY(M).EQ.0) GO TO 10                                           00000740
      I=IJS(M)                                                             00000750
      SMF=SMF+TX(I)*PHIX(I)                                                00000760
      SMS=SMS+TTH(I)*PHIT(I)                                               00000770
      SMV=SMV-PHIT(I)*TXT(I)                                               00000780
      SME=SME+PHIX(I)*TXT(I)                                               00000790
      SMN=SMN+TX(I)*PHI(I)                                                 00000800
      SMT=SMT+TTH(I)*PHI(I)                                                00000810
      GO TO 10                                                             00000820
   20 DO 21 L=1,MNMAXO                                                     00000830
      SMF=SMF+TX(L)*PHIX(L)                                                00000840
   21 SMV=SMV+PHIT(L)*TXT(L)                                               00000850
```

```
      IF(M.GT.MNMAXO) GO TO 10                                          00000860
      SMF=SMF+TX(M)*PHIX(M)                                             00000870
   10 EXX3(M)=SMF*.5                                                    00000880
      ETT3(M)=SMS*.5                                                    00000890
      ETX3(M)=SMV*.5                                                    00000900
      EXT3(M)=SME*.5                                                    00000910
      EX3(M)=SMN*.5                                                     00000920
      ET3(M)=SMT*.5                                                     00000930
    9 CONTINUE                                                          00000940
      DO 30 M=1,MNMAXO                                                  00000950
      U1(M)=U2(M)                                                       00000960
      V1(M)=V2(M)                                                       00000970
      W1(M)=W2(M)                                                       00000980
      U2(M)=U3(M)                                                       00000990
      V2(M)=V3(M)                                                       00001000
   30 W2(M)=W3(M)                                                       00001010
      RETURN                                                            00001020
      END                                                               00001030
```

APPENDIX F

LISTING OF SATANS-II MODIFIED SUBROUTINES AND COMMON STATEMENTS

AS INDICATED IN THE BODY OF THIS REPORT, REPLACE THE FOLLOWING COMMON STATEMENTS IN ALL SUBROUTINES WHERE THEY OCCUR.

```
      COMMON /IBL2/  N(25),MNINIT
      COMMON /IBL7/  MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),
     1               JS(25,25),ID(25,25),JD(25,25),IJS(25)
      COMMON /IBL11/  ICORFL,IPASS,JUMP,MPERFS
      COMMON /BL3/   PR(25),PX(25),PT(25)
      COMMON /BL4/   P(4,4,951),X(4,951),ZF1M(4,4,25),ZF2M(4,4,25),
     1               ZF3M(4,4,25),ZF4M(4,4,25)
      COMMON /BL5/   TT(25),MT(25),DT(25),DMT(25)
      COMMON /BL6/   Z(4,1001),SOE,OSE,ALOAD
      COMMON /BL8/   R(100),GAM(100),OMT(100)
      COMMON /BL9/   FFS(4,25),EL1S(4),GEES(4,25)
      COMMON /BL10/  PHIX(25),PHIT(25),PHI(25)
      COMMON /BL11/  OMXI(100),PHEE,T0,T2
      COMMON /BL15/  NU,U1(25),V1(25),W1(25),V2(25),U2(25),W2(25),U3(25),
     1               V3(25),W3(25)
      COMMON /BL20/  DEOMX(100)
      COMMON /BL24/  DL(4,4,25),DG(4,4,25),DF(4,4,25)
      COMMON /BL27/  BX3(25),BT3(25),BXT3(25),BE3(25)
      COMMON /BL28/  EXX3(25),ETT3(25),ETX3(25),EXT3(25),EX3(25),ET3(25)
      COMMON /BL29/  BX1(25),BT1(25),BXT1(25),BE1(25),BX2(25),BT2(25),
     1               BXT2(25),BE2(25)
      COMMON /BL30/  EXX1(25),ETT1(25),ETX1(25),EX1(25),ET1(25),EXX2(25),
     1               ETT2(25),ETX2(25),EXT2(25),EX2(25),ET2(25)
      COMMON /BL31/  DELSQ,EXT1(25)
      COMMON /BL34/  DEE(4,4,951),DST(4,4,951)
      COMMON /BL101/  Z0(4,1001),Z2(4,1001),Z3(4,1001),DELSD
      COMMON /BL103/  MASS(100)
      COMMON /BL104/  ZDOT(4,1001)
      COMMON /BL110/  TX(25),TTH(25),TXT(25),MX(25),MTH(25),MXT(25),
     1               QS(25)
```

AS INDICATED IN THE BODY OF THIS REPORT, REPLACE SUBROUTINES TEAETA, PHIBET, MODES, OUTPUT, INLPOL, FNLPOL AND POLE IN APPENDIX E WITH THOSE THAT FOLLOW.

```
      SUBROUTINE TEAETA(K)                                                  00000010
C*****************************************************************        00000020
C         THIS SUBROUTINE CALCULATES THE INPLANE FORCES AND CARRIES OUT *   00000030
C         THE MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE AETA*   00000040
C         NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K. THE ARRAYS *   00000050
C         'IS','JS','ID','JD','IJS','MAXS','MAXD', AND 'MAXSY' PREPARED *  00000060
C         IN SUBROUTINE MODES ARE USED HERE                             *   00000070
C*****************************************************************        00000080
      REAL NU,MT                                                            00000090
      COMMON /IBL1/ MNMAX                                                   00000100
      COMMON /IBL2/ N(25),MNINIT                                            00000110
      COMMON /IBL7/ MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),           00000120
     1              JS(25,25),ID(25,25),JD(25,25),IJS(25)                   00000130
      COMMON /IBL8/ LSTEP,ITR                                               00000140
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                               00000150
      COMMON /IBL13/ ITRMAX,LSMAX                                           00000160
      COMMON /BL5/  TT(25),MT(25),DT(25),DMT(25)                            00000170
      COMMON /BL6/  Z(4,1001),SOE,OSE,ALOAD                                 00000180
      COMMON /BL7/  D1,S1                                                   00000190
      COMMON /BL8/  R(100),GAM(100),OMT(100)                                00000200
      COMMON /BL10/ PHIX(25),PHIT(25),PHI(25)                               00000210
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                    00000220
      COMMON /BL12/ TDLI,TDEL                                               00000230
      COMMON /BL15/ NU,U1(25),U2(25),U3(25),V1(25),V2(25),V3(25),W1(25),    00000240
     1              W2(25),W3(25)                                           00000250
      COMMON /BL27/ BX3(25),BT3(25),BXT3(25),BE3(25)                        00000260
      COMMON /BL28/ EXX3(25),ETT3(25),ETX3(25),EXT3(25),EX3(25),ET3(25)     00000270
      COMMON /BLPHS/ PHX(25),PHT(25)                                        00000280
C*****************************************************************        00000281
      DIMENSION TX(25),TTH(25),TXT(25)                                      00000290
      RRA=1./R(K)                                                           00000300
      GA=GAM(K)                                                             00000310
      OX=OMXI(K)                                                            00000320
      OT=OMT(K)                                                             00000330
      CALL BDB(K,BS,DB,DS,DD)                                               00000340
      DO 1 M=1,MNMAXO                                                       00000350
      PHIX(M)=PHIX(M)+PHX(M)                                                00000360
      PHIT(M)=PHIT(M)+PHT(M)                                                00000370
      EN=N(M)                                                               00000380
      CALL TLOAD(K)                                                         00000390
```

```
      TTS=TT(M)*ALOAD                                                    00000400
      EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+OSE*(BX3(M)+BE3(M))                 00000410
      ET=EN*V2(M)*RRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M))              00000420
      EXT=.5*(TDLI*(V3(M)-V1(M))- EN *U2(M)*RRA-GA*V2(M))                00000430
     1     +OSE*BXT3(M)                                                 00000440
      TX(M)=BS*(EX+NU*ET)-TTS                                            00000450
      TTH(M)=BS*(ET+NU*EX)-TTS                                           00000460
     1 TXT(M)=BS*D1*EXT                                                  00000470
      IF (JUMP.EQ.2) GO TO 1111                                          00000480
C*************************************************************************00000490
C                                                                       *00000500
C                                                                       *00000510
C     THIS SECTION HANDLES SINGLE-TERM EXPANSIONS ONLY                  *00000520
C                                                                       *00000530
C                                                                       *00000540
C*************************************************************************00000550
      DO 19 M=1,MNMAX                                                    00000560
      SMF=0.0                                                            00000570
      SMS=0.0                                                            00000580
      SMV=0.0                                                            00000590
      SME=0.0                                                            00000600
      SMN=0.0                                                            00000610
      SMT=0.0                                                            00000620
      IF(N(M).EQ.0) GO TO 120                                            00000630
      MAXL=MAXS(M)                                                       00000640
      IF(MAXL.EQ.0) GO TO 12                                             00000650
      DO 13 L=1,MAXL                                                     00000660
      I=IS(L,M)                                                          00000670
      J=JS(L,M)                                                          00000680
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                00000690
      SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                              00000700
      SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)                              00000710
      SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                              00000720
      SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)                                  00000730
   13 SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)                                00000740
   12 MAXL=MAXD(M)                                                       00000750
      IF(MAXL.EQ.0) GO TO 14                                             00000760
      DO 15 L=1,MAXL                                                     00000770
      I=ID(L,M)                                                          00000780
      J=JD(L,M)                                                          00000790
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                00000800
      SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                              00000810
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)                              00000820
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                              00000830
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)                                  00000840
   15 SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)                                00000850
   14 IF(MAXSY(M).EQ.0) GO TO 110                                        00000860
      I=IJS(M)                                                           00000870
```

```
      SMF=SMF+TX(I)*PHIX(I)                                                    00000880
      SMS=SMS+TTH(I)*PHIT(I)                                                   00000890
      SMV=SMV-PHIT(I)*TXT(I)                                                   00000900
      SME=SME+PHIX(I)*TXT(I)                                                   00000910
      SMN=SMN+TX(I)*PHI(I)                                                     00000920
      SMT=SMT+TTH(I)*PHI(I)                                                    00000930
      GO TO 110                                                                00000940
  120 DO 121 L=1,MNMAXO                                                        00000950
      SMF=SMF+TX(L)*PHIX(L)                                                    00000960
  121 SMV=SMV+PHIT(L)*TXT(L)                                                   00000970
      IF(M.GT.MNMAXO) GO TO 110                                                00000980
      SMF=SMF+TX(M)*PHIX(M)                                                    00000990
  110 EXX3(M)=SMF*.5                                                           00001000
      ETT3(M)=SMS*.5                                                           00001010
      ETX3(M)=SMV*.5                                                           00001020
      EXT3(M)=SME*.5                                                           00001030
      EX3(M)=SMN*.5                                                            00001040
      ET3(M)=SMT*.5                                                            00001050
   19 CONTINUE                                                                 00001060
      GO TO 2222                                                               00001070
C*************************************************************************00001080
C                                                                        *00001090
C                                                                        *00001100
C     THIS SECTION HANDLES GENERAL LOADING                               *00001110
C                                                                        *00001120
C                                                                        *00001130
C*************************************************************************00001140
 1111 DO 9 M=1,MNMAX                                                         00001150
      SMF=0.0                                                                  00001160
      SMS=0.0                                                                  00001170
      SMV=0.0                                                                  00001180
      SME=0.0                                                                  00001190
      SMN=0.0                                                                  00001200
      SMT=0.0                                                                  00001210
C*************************************************************************00001220
C     TEST FOR ASYMMETRIC MODE                                           *00001230
C*************************************************************************00001240
      IF (N(M).LT.0) GO TO 101                                                 00001250
C*************************************************************************00001260
C                                                                        *00001270
C     THIS SECTION HANDLES SYMMETRIC COMBINATIONS                        *00001280
C                                                                        *00001290
C*************************************************************************00001300
C     TEST FOR ZERO-TH MODE                                              *00001310
C*************************************************************************00001320
      IF (N(M).EQ.0) GO TO 20                                                  00001330
      MAXL=MAXS(M)                                                             00001340
C*************************************************************************00001350
```

```
C      TEST FOR PRESENCE OF SYMMETRIC SUMMATION COMBINATIONS                *00001360
C*****************************************************************************00001370
       IF (MAXL.EQ.0) GO TO 2                                                 00001380
C*****************************************************************************00001390
C      SET UP COUPLING MODES' INDICES AND TEST FOR MODE 1                   *00001400
C*****************************************************************************00001410
       DO 3 L=1,MAXL                                                          00001420
       I=IS(L,M)                                                              00001430
       J=JS(L,M)                                                              00001440
       II=I-1                                                                 00001450
       JJ=J-1                                                                 00001460
       IF (I.EQ.1) GO TO 3                                                    00001470
C*****************************************************************************00001480
C      COMPILE SUMS FOR SYMMETRIC SUM COMBINATION MODES                     *00001490
C*****************************************************************************00001500
       SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)-PHIX(II)*TX(JJ)                    00001510
     1        -PHIX(JJ)*TX(II)                                                00001520
       SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)+PHIT(II)*TTH(JJ)                 00001530
     1        +PHIT(JJ)*TTH(II)                                               00001540
       SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)+PHIT(II)*TXT(JJ)                 00001550
     1        +PHIT(JJ)*TXT(II)                                               00001560
       SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)+PHIX(II)*TXT(JJ)                 00001570
     1        +PHIX(JJ)*TXT(II)                                               00001580
       SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)+PHI(II)*TX(JJ)                       00001590
     1        +PHI(JJ)*TX(II)                                                 00001600
       SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)+PHI(II)*TTH(JJ)                    00001610
     1        +PHI(JJ)*TTH(II)                                                00001620
     3 CONTINUE                                                               00001630
     2 MAXL=MAXD(M)                                                           00001640
C*****************************************************************************00001650
C      TEST FOR PRESENCE OF SYMMETRIC DIFFERENCE COMBINATIONS               *00001660
C*****************************************************************************00001670
       IF (MAXL.EQ.0) GO TO 4                                                 00001680
C*****************************************************************************00001690
C       SET UP COUPLING MODES' INDICES AND TEST FOR MODE 1                  *00001700
C*****************************************************************************00001710
       DO 5 L=1,MAXL                                                          00001720
       I=ID(L,M)                                                              00001730
       J=JD(L,M)                                                              00001740
       II=I-1                                                                 00001750
       JJ=J-1                                                                 00001760
       IF (J.EQ.1) GO TO 42                                                   00001770
C*****************************************************************************00001780
C      COMPILE SUMS FOR SYMMETRIC DIFFERENCE COMBINATIONS                   *00001790
C*****************************************************************************00001800
       SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)+PHIX(II)*TX(JJ)                    00001810
     1        +PHIX(JJ)*TX(II)                                                00001820
       SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)-PHIT(II)*TTH(JJ)                 00001830
```

```
     1        +PHIT(JJ)*TTH(II)                                              00001840
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)+PHIT(II)*TXT(JJ)                00001850
     1        +PHIT(JJ)*TXT(II)                                              00001860
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)+PHIX(II)*TXT(JJ)                00001870
     1        -PHIX(JJ)*TXT(II)                                              00001880
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)-PHI(II)*TX(JJ)                      00001890
     1        +PHI(JJ)*TX(II)                                                00001900
      SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)-PHI(II)*TTH(JJ)                   00001910
     1        +PHI(JJ)*TTH(II)                                               00001920
      GO TO 5                                                               00001930
C*************************************************************************00001940
C     EXECUTE BELOW IF J=1 IN DIFF-COMB. (OR I=1 IN SUM COMB)            *00001950
C*************************************************************************00001960
   42 SMF=SMF+(PHIX(1)*TX(I)+PHIX(I)*TX(1))*2.0                             00001970
      SMS=SMS+(PHIT(I)*TTH(1)+PHIT(1)*TTH(II))*2.0                          00001980
      SMV=SMV+(PHIT(1)*TXT(II)+PHIT(II)*TXT(1))*2.0                         00001990
      SME=SME+(PHIX(II)*TXT(1)+PHIX(1)*TXT(II))*2.0                         00002000
      SMN=SMN+(PHI(I)*TX(1)+PHI(1)*TX(II))*2.0                              00002010
      SMT=SMT+(PHI(I)*TTH(1)+PHI(1)*TTH(II))*2.0                            00002020
    5 CONTINUE                                                              00002030
C*************************************************************************00002040
C     TEST FOR PRESENCE OF SAME-INDEX COMBINATIONS                       *00002050
C*************************************************************************00002060
    4 IF (MAXSY(M).EQ.0) GO TO 10                                           00002070
C*************************************************************************00002080
C     SET UP INDICES AND COMPILE SUMS                                    *00002090
C*************************************************************************00002100
      I=IJS(M)                                                              00002110
      II=I-1                                                                00002120
      SMF=SMF+TX(I)*PHIX(I)-PHIX(II)*TX(II)                                 00002130
      SMS=SMS+TTH(I)*PHIT(I)+PHIT(II)*TTH(II)                               00002140
      SMV=SMV-PHIT(I)*TXT(I)+PHIT(II)*TXT(II)                               00002150
      SME=SME+PHIX(I)*TXT(I)+PHIX(II)*TXT(II)                               00002160
      SMN=SMN+TX(I)*PHI(I)+PHI(II)*TX(II)                                   00002170
      SMT=SMT+TTH(I)*PHI(I)+PHI(II)*TTH(II)                                 00002180
      GO TO 10                                                              00002190
C*************************************************************************00002200
C     HERE WE HANDLE CASES WHERE N(M)=0                                  *00002210
C*************************************************************************00002220
   20 DO 21 L=1,MNMAXO                                                      00002230
      SMF=SMF+TX(L)*PHIX(L)                                                 00002240
      SMV=SMV+TXT(L)*PHIT(L)                                                00002250
   21 CONTINUE                                                              00002260
      DO 22 L=3,MNMAXO,2                                                    00002270
      LL=L-1                                                                00002280
      SMS=SMS+PHIT(LL)*TTH(L)+PHIT(L)*TTH(LL)                               00002290
      SME=SME+PHIX(L)*TXT(LL)+PHIX(LL)*TXT(L)                               00002300
      SMN=SMN+PHI(LL)*TX(L)+PHI(L)*TX(LL)                                   00002310
```

```
      SMT=SMT+PHI(LL)*TTH(L)+PHI(L)*TTH(LL)                              00002320
   22 CONTINUE                                                           00002330
      SMF=SMF+TX(1)*PHIX(1)                                              00002340
      SMS=SMS+TTH(1)*PHIT(1)*2.0                                         00002350
      SMV=SMV+TXT(1)*PHIT(1)                                             00002360
      SME=SME+TXT(1)*PHIX(1)*2.0                                         00002370
      SMN=SMN+TX(1)*PHI(1)*2.0                                           00002380
      SMT=SMT+TTH(1)*PHI(1)*2.0                                          00002390
      GO TO 10                                                           00002400
C*************************************************************************00002410
C                                                                        *00002420
C     THIS SECTION HANDLES ASYMMETRIC MODE COMBINATIONS                  *00002430
C                                                                        *00002440
C*************************************************************************00002450
  101 MP=M+1                                                             00002460
      MAXL=MAXS(MP)                                                      00002470
C*************************************************************************00002480
C     TEST FOR PRESENCE OF SUMMATION COMBINATIONS                        *00002490
C*************************************************************************00002500
      IF (MAXL.EQ.0) GO TO 102                                           00002510
C*************************************************************************00002520
C     SET UP COUPLING INDICES AND TEST FOR MODE 1                        *00002530
C*************************************************************************00002540
      DO 103 L=1,MAXL                                                    00002550
      I=IS(L,MP)                                                         00002560
      J=JS(L,MP)                                                         00002570
      II=I-1                                                             00002580
      JJ=J-1                                                             00002590
      IF (I.EQ.1) GO TO 103                                              00002600
C*************************************************************************00002610
C     COMPILE SUMS FOR ASYMMETRIC SUMMATION COMBINATIONS                 *00002620
C*************************************************************************00002630
      SMF=SMF+PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)               00002640
     1       +PHIX(JJ)*TX(I)                                             00002650
      SMS=SMS-PHIT(I)*TTH(JJ)-PHIT(J)*TTH(II)+PHIT(II)*TTH(J)            00002660
     1       +PHIT(JJ)*TTH(I)                                            00002670
      SMV=SMV+PHIT(I)*TXT(JJ)+PHIT(J)*TXT(II)+PHIT(II)*TXT(J)            00002680
     1       +PHIT(JJ)*TXT(I)                                            00002690
      SME=SME+PHIX(I)*TXT(JJ)+PHIX(J)*TXT(II)-PHIX(II)*TXT(J)            00002700
     1       -PHIX(JJ)*TXT(I)                                            00002710
      SMN=SMN-PHI(I)*TX(JJ)-PHI(J)*TX(II)+PHI(II)*TX(J)                  00002720
     1       +PHI(JJ)*TX(I)                                              00002730
      SMT=SMT-PHI(I)*TTH(JJ)-PHI(J)*TTH(II)+PHI(II)*TTH(J)               00002740
     1       +PHI(JJ)*TTH(I)                                             00002750
  103 CONTINUE                                                           00002760
C*************************************************************************00002770
C     TEST FOR PRESENCE OF DIFFERENCE COMBINATIONS                       *00002780
C*************************************************************************00002790
```

```
  102 MAXL=MAXD(MP)                                                      00002800
      IF (MAXL.EQ.0) GO TO 104                                           00002810
C*****************************************************************     00002820
C     SET UP COUPLING MODES' INDICES AND TEST FOR MODE 1               *00002830
C*****************************************************************     00002840
      DO 105 L=1,MAXL                                                    00002850
      I=ID(L,MP)                                                         00002860
      J=JD(L,MP)                                                         00002870
      II=I-1                                                             00002880
      JJ=J-1                                                             00002890
      IF (J.EQ.1) GO TO 123                                              00002900
C*****************************************************************     00002910
C     COMPILE SUMS FOR ASYMMETRIC DIFFERENCE COMBINATIONS              *00002920
C*****************************************************************     00002930
      SMF=SMF-PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)               00002940
     1       -PHIX(JJ)*TX(I)                                             00002950
      SMS=SMS+PHIT(I)*TTH(JJ)+PHIT(J)*TTH(II)+PHIT(II)*TTH(J)            00002960
     1       +PHIT(JJ)*TTH(I)                                            00002970
      SMV=SMV+PHIT(I)*TXT(JJ)+PHIT(J)*TXT(II)+PHIT(II)*TXT(J)            00002980
     1       +PHIT(JJ)*TXT(I)                                            00002990
      SME=SME+PHIX(I)*TXT(JJ)+PHIX(J)*TXT(II)+PHIX(II)*TXT(J)            00003000
     1       +PHIX(JJ)*TXT(I)                                            00003010
      SMN=SMN+PHI(I)*TX(JJ)+PHI(J)*TX(II)+PHI(II)*TX(J)                  00003020
     1       +PHI(JJ)*TX(I)                                              00003030
      SMT=SMT+PHI(I)*TTH(JJ)+PHI(J)*TTH(II)+PHI(II)*TTH(J)               00003040
     1       +PHI(JJ)*TTH(I)                                             00003050
      GO TO 105                                                          00003060
C*****************************************************************     00003070
C     EXECUTE BELOW IF J=1 IN DIFF-COMB (OR I=1 IN SUM-COMB)           *00003080
C*****************************************************************     00003090
  123 SMF=SMF+(PHIX(1)*TX(II)+PHIX(II)*TX(1))*2.0                        00003100
      SMS=SMS+(PHIT(1)*TTH(I)+PHIT(II)*TTH(1))*2.0                       00003110
      SMV=SMV+(PHIT(I)*TXT(1)+PHIT(1)*TXT(I))*2.0                        00003120
      SME=SME+(PHIX(1)*TXT(II)+PHIX(I)*TXT(1))*2.0                       00003130
      SMN=SMN+(PHI(1)*TX(I)+PHI(II)*TX(1))*2.0                           00003140
      SMT=SMT+(PHI(1)*TTH(I)+PHI(II)*TTH(1))*2.0                         00003150
  105 CONTINUE                                                           00003160
C*****************************************************************     00003170
C     TEXT FOR PRESENCE OF SAME-INDEX COMBINATION                      *00003180
C*****************************************************************     00003190
  104 IF (MAXSY(MP).EQ.0) GO TO 10                                       00003200
C*****************************************************************     00003210
C     SET UP COUPLING MODES' INDICES AND COMPILE SUMS                  *00003220
C*****************************************************************     00003230
      I=IJS(MP)                                                          00003240
      II=I-1                                                             00003250
      SMF=SMF+PHIX(I)*TX(II)+PHIX(II)*TX(I)                              00003260
      SMS=SMS-PHIT(I)*TTH(II)+PHIT(II)*TTH(I)                            00003270
```

```
      SMV=SMV+PHIT(I)*TXT(II)+PHIT(II)*TXT(I)                          00003280
      SME=SME+PHIX(I)*TXT(II)-PHIX(II)*TXT(I)                          00003290
      SMN=SMN-PHI(I)*TX(II)+PHI(II)*TX(I)                              00003300
      SMT=SMT-PHI(I)*TTH(II)+PHI(II)*TTH(I)                            00003310
   10 CONTINUE                                                         00003320
C*****************************************************************00003330
C     PREPARE 'AETA' TERMS                                           *00003340
C*****************************************************************00003350
      EXX3(M)=SMF*0.5                                                  00003360
      ETT3(M)=SMS*0.5                                                  00003370
      ETX3(M)=SMV*0.5                                                  00003380
      EXT3(M)=SME*0.5                                                  00003390
      EX3(M) =SMN*0.5                                                  00003400
      ET3(M) =SMT*0.5                                                  00003410
    9 CONTINUE                                                         00003420
 2222 CONTINUE                                                         00003430
C*****************************************************************00003440
C     UP-DATE U,V,W AND RETURN                                       *00003450
C*****************************************************************00003460
      DO 30 M=1,MNMAXO                                                 00003470
      U1(M)=U2(M)                                                      00003480
      V1(M)=V2(M)                                                      00003490
      W1(M)=W2(M)                                                      00003500
      U2(M)=U3(M)                                                      00003510
      V2(M)=V3(M)                                                      00003520
   30 W2(M)=W3(M)                                                      00003530
      RETURN                                                           00003540
      END                                                              00003550
```

```
      SUBROUTINE PHIBET(K)                                             00000010
C*****************************************************************00000020
C         THIS SUBROUTINE CALCULATES THE PHI'S AND CARRIES OUT THE     *00000030
C         MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE BETA   *00000040
C         NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K. THE ARRAYS *00000050
C         IS, JS, ID, JS,IJS, MAXS, MAXD, MAXSY ARE PREPARED IN SUB-   *00000060
C         ROUTINE MODES AND USED HERE.                                 *00000070
C*****************************************************************00000080
```

```
      COMMON /IBL1/ MNMAX                                                    00000090
      COMMON /IBL2/ N(25),MNINIT                                             00000100
      COMMON /IBL4/ KMAX,KL                                                  00000110
      COMMON /IBL7/ MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),            00000120
     1              JS(25,25),ID(25,25),JD(25,25),IJS(25)                   00000130
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                                00000140
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                       00000150
      COMMON /IBL13/ ITRMAX,LSMAX                                            00000160
      COMMON /BL6/  Z(4,1001),SOE,OSE,ALOAD                                  00000170
      COMMON /BL8/  R(100),GAM(100),OMT(100)                                 00000180
      COMMON /BL10/ PHIX(25),PHIT(25),PHI(25)                                00000190
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                     00000200
      COMMON /BL12/ TDLI,TDEL                                                00000210
      COMMON /BL15/ NU,U1(25),U2(25),U3(25),V1(25),V2(25),V3(25),W1(25),     00000220
     1              W2(25),W3(25)                                            00000230
      COMMON /BL27/ BX3(25),BT3(25),BXT3(25),BE3(25)                         00000240
      COMMON /BLIMP/ PHIXB(951),PHITB(951)                                   00000250
      COMMON /BLPHS/ PHX(25),PHT(25)                                         00000260
C*************************************************************************00000261
      OX=OMXI(K)                                                             00000270
      OT=OMT(K)                                                              00000280
      RRA=1./R(K)                                                            00000290
      GA=GAM(K)                                                              00000300
      KP2=K+2                                                                00000310
      DO 1 M=1,MNMAXO                                                        00000320
      EN=N(M)                                                                00000330
      IK=KP2+(M-1)*<MAX2                                                     00000340
      U3(M)=Z(1,IK)                                                          00000350
      V3(M)=Z(2,IK)                                                          00000360
      W3(M)=Z(3,IK)                                                          00000370
      PHIX(M)=-TDLI*(W3(M)-W1(M))+OX*U2(M)                                   00000380
      PHIT(M)=EN*W2(M)*RRA+V2(M)*OT                                          00000390
    1 PHI(M)=(TDLI*(V3(M)-V1(M))+GA*V2(M)+EN*U2(M)*RRA)*.5                   00000400
      IF(ITRMAX.EQ.1) RETURN                                                 00000410
      IF (JUMP.EQ.2) GO TO 1111                                              00000420
C*************************************************************************00000430
C                                                                           *00000440
C                                                                           *00000450
C     THIS SECTION HANDLES  'SINGLE-TERM' EXPANSIONS ONLY                   *00000460
C                                                                           *00000470
C                                                                           *00000480
C*************************************************************************00000490
      DO 19 M=1,MNMAX                                                        00000500
      SMO=0.                                                                 00000510
      SMT=0.                                                                 00000520
      SMR=0.                                                                 00000530
      SMF=0.                                                                 00000540
      IF(N(M).EQ.0) GO TO 120                                                00000550
```

```
      MAXL=MAXS(M)                                                      00000560
      IF(MAXL.EQ.0) GO TO 12                                            00000570
      DO 13 L=1,MAXL                                                    00000580
      I=IS(L,M)                                                         00000590
      J=JS(L,M)                                                         00000600
      SMO=SMO+PHIX(I)*PHIX(J)                                           00000610
      SMT=SMT-PHIT(I)*PHIT(J)                                           00000620
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                           00000630
   13 SMF=SMF-PHI(I)*PHI(J)                                             00000640
   12 MAXL=MAXD(M)                                                      00000650
      IF(MAXL.EQ.0) GO TO 14                                            00000660
      DO 15 L=1,MAXL                                                    00000670
      I=ID(L,M)                                                         00000680
      J=JD(L,M)                                                         00000690
      SMO=SMO+PHIX(I)*PHIX(J)                                           00000700
      SMT=SMT+PHIT(I)*PHIT(J)                                           00000710
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                           00000720
   15 SMF=SMF+PHI(I)*PHI(J)                                             00000730
   14 IF(MAXSY(M).EQ.0) GO TO 110                                       00000740
      I=IJS(M)                                                          00000750
      SMO=SMO+PHIX(I)**2/2.                                             00000760
      SMT=SMT-PHIT(I)**2/2.                                             00000770
      SMR=(SMR+PHIX(I)*PHIT(I))                                         00000780
      SMF=SMF-PHI(I)**2/2.                                              00000790
      GO TO 110                                                         00000800
  120 DO 222 L=1,MNMAXO                                                 00000810
      SMO=SMO+PHIX(L)**2                                                00000820
      SMT=SMT+PHIT(L)**2                                                00000830
  222 SMF=SMF+PHI(L)**2                                                 00000840
      IF(M.GT.MNMAXO) GO TO 111                                         00000850
      SMO=SMO+PHIX(M)**2                                                00000860
  111 BX3(M)=SMO*.5                                                     00000870
      BT3(M)=SMT*.5                                                     00000880
      BE3(M)=SMF*.5                                                     00000890
      BXT3(M)=0.                                                        00000900
      GO TO 19                                                          00000910
  110 BX3(M)=SMO                                                        00000920
      BT3(M)=SMT                                                        00000930
      BXT3(M)=SMR*.5                                                    00000940
      BE3(M)=SMF                                                        00000950
   19 CONTINUE                                                          00000960
      GO TO 2222                                                        00000970
C*****************************************************************    00000980
C                                                                     *  00000990
C                                                                     *  00001000
C     THIS SECTION HANDLES GENERAL LOADING AND IMPERFECTIONS          *  00001010
C                                                                     *  00001020
C                                                                     *  00001030
```

```
C*****************************************************************00001040
 1111 DO 60 M=1,MNMAXO                                               00001050
      KP=K+(M-1)*KMAX                                                00001060
      PHX(M)=PHIXB(KP)                                               00001070
   60 PHT(M)=PHITB(KP)                                               00001080
      DO 9 M=1,MNMAX                                                 00001090
      SMO=0.                                                         00001100
      SMT=0.                                                         00001110
      SMR=0.                                                         00001120
      SMF=0.                                                         00001130
C*****************************************************************00001140
C     TEST FOR ASYMMETRIC MODE                                       *00001150
C*****************************************************************00001160
      IF (N(M).LT.0) GO TO 101                                       00001170
C*****************************************************************00001180
C                                                                    *00001190
C     THIS SECTION HAMDLES SYMMETRIC MODE COMBINATIONS ONLY           *00001200
C                                                                    *00001210
C*****************************************************************00001220
C     TEST FOR ZERO-TH MODE                                          *00001230
C*****************************************************************00001240
      IF(N(M).EQ.0) GO TO 20                                         00001250
      MAXL=MAXS(M)                                                   00001260
C*****************************************************************00001270
C     TEST FOR PRESENCE OF SYMMETRIC SUMMATION COMBINATIONS          *00001280
C*****************************************************************00001290
      IF(MAXL.EQ.0) GO TO 2                                          00001300
C*****************************************************************00001310
C     SET UP 'COUPLING' MODES' INDICES AND TEST FOR MODE 1           00001320
C*****************************************************************00001330
      DO 3 L=1,MAXL                                                  00001340
      I=IS(L,M)                                                      00001350
      J=JS(L,M)                                                      00001360
      II=I-1                                                         00001370
      JJ=J-1                                                         00001380
      IF (I.EQ.1) GO TO 3                                            00001390
C*****************************************************************00001400
C     COMPILE SUMS FOR SYMMETRIC SUMMATION COMBINATION MODES          *00001410
C*****************************************************************00001420
      SMO=SMO+PHIX(I)*PHIX(J)                                        00001430
     1     +PHX(I)*PHIX(J)+PHX(J)*PHIX(I)                            00001440
     2          -PHIX(II)*(PHIX(JJ)+PHX(JJ))-PHIX(JJ)*PHX(II)        00001450
      SMT=SMT-PHIT(I)*PHIT(J)                                        00001460
     1     -PHT(I)*PHIT(J)-PHT(J)*PHIT(I)                            00001470
     2          +PHIT(II)*(PHIT(JJ)+PHT(JJ))+PHIT(JJ)*PHT(II)        00001480
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                        00001490
     1     +PHX(I)*PHIT(J)+PHX(J)*PHIT(I)+PHIX(I)*PHT(J)             00001500
     2     +PHIX(J)*PHT(I)                                           00001510
```

```
     3          +PHIX(II)*(PHIT(JJ)+PHT(JJ))+PHIX(JJ)*(PHIT(II)                00001520
     4          +PHT(II))+PHX(II)*PHIT(JJ)+PHX(JJ)*PHIT(II)                    00001530
      SMF=SMF-PHI(I)*PHI(J)                                                    00001540
     1          +PHI(II)*PHI(JJ)                                              00001550
    3 CONTINUE                                                                 00001560
    2 MAXL=MAXD(M)                                                             00001570
C*************************************************************************** 00001580
C       TEST FOR PRESENCE OF SYMMETRIC DIFFERENCE COMBINATIONS                *00001590
C*************************************************************************** 00001600
      IF(MAXL.EQ.0) GO TO 4                                                    00001610
C*************************************************************************** 00001620
C      SET UP 'COUPLING' MODES' INDICES AND TEST FOR MODE 1                   *00001630
C*************************************************************************** 00001640
      DO 5 L=1,MAXL                                                            00001650
      I=ID(L,M)                                                                00001660
      J=JD(L,M)                                                                00001670
      II=I-1                                                                   00001680
      JJ=J-1                                                                   00001690
      IF (J.EQ.1) GO TO 42                                                     00001700
C*************************************************************************** 00001710
C      COMPILE SUMS FOR SYMMETRIC DIFFERENCE COMBINATION MODES                *00001720
C*************************************************************************** 00001730
      SMO=SMO+PHIX(I)*PHIX(J)                                                  00001740
     1     +PHX(I)*PHIX(J)+PHX(J)*PHIX(I)                                      00001750
     2          +PHIX(II)*(PHIX(JJ)+PHX(JJ))+PHIX(JJ)*PHX(II)                 00001760
      SMT=SMT+PHIT(I)*PHIT(J)                                                  00001770
     1     +PHT(I)*PHIT(J)+PHT(J)*PHIT(I)                                      00001780
     2          +PHIT(II)*(PHIT(JJ)+PHT(JJ))+PHIT(JJ)*PHT(II)                 00001790
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                                  00001800
     1     -PHX(I)*PHIT(J)+PHX(J)*PHIT(I)-PHIX(I)*PHT(J)                       00001810
     2     +PHIX(J)*PHT(I)                                                     00001820
     3          +PHIX(II)*(PHIT(JJ)+PHT(JJ))-PHIX(JJ)*(PHIT(II)               00001830
     4          +PHT(II))+PHX(II)*PHIT(JJ)-PHX(JJ)*PHIT(II)                   00001840
      SMF=SMF+PHI(I)*PHI(J)                                                    00001850
     1          +PHI(II)*PHI(JJ)                                              00001860
      GO TO 5                                                                  00001870
C*************************************************************************** 00001880
C      EXECUTE BELOW IF J=1 IN DIFFERENCE COMBINATIONS                        *00001890
C*************************************************************************** 00001900
   42 SMO=SMO+(PHIX(1)*PHIX(I)+PHX(1)*PHIX(I)+PHX(I)*PHIX(1))*2.0              00001910
      SMT=SMT+(PHIT(1)*(PHIT(II)+PHT(II))+PHIT(II)*PHT(1))*2.0                 00001920
      SMR=SMR+(PHIX(1)*(PHIT(I)+PHT(I))+PHX(1)*PHIT(I)                         00001930
     1     +PHIT(1)*(PHIX(II)+PHX(II))+PHT(1)*PHIX(II))*2.0                    00001940
      SMF=SMF+(PHI(1)*PHI(II))*2.0                                             00001950
    5 CONTINUE                                                                 00001960
C*************************************************************************** 00001970
C      TEST FOR PRESENCE OF SAME-INDEX COMBINATION                            *00001980
C*************************************************************************** 00001990
```

```
    4 IF(MAXSY(M).EQ.0) GO TO 10                                          00002000
C*****************************************************************************00002010
C     SET UP INDICES AND COMPILE SUMS                                     *00002020
C*****************************************************************************00002030
      I=IJS(M)                                                            00002040
      II=I-1                                                              00002050
      SMO=SMO+PHIX(I)**2/2.                                               00002060
     1     +PHX(I)*PHIX(I)                                                00002070
     2          -PHIX(II)**2/2.0-PHIX(II)*PHX(II)                         00002080
      SMT=SMT-PHIT(I)**2/2.                                               00002090
     1     -PHT(I)*PHIT(I)                                                00002100
     2          +PHIT(II)**2/2.0+PHIT(II)*PHT(II)                         00002110
      SMR=(SMR+PHIX(I)*PHIT(I))                                           00002120
     1     +PHX(I)*PHIT(I)+PHIX(I)*PHT(I)                                 00002130
     2          +PHIX(II)*(PHIT(II)+PHT(II))+PHX(II)*PHIT(II)             00002140
      SMF=SMF-PHI(I)**2/2.                                                00002150
     2          +PHI(II)**2/2.0                                           00002160
      GO TO 10                                                            00002170
C*****************************************************************************00002180
C     THIS SECTION HANDLES CASES WHERE MODE 0 IS INCLUDED                 *00002190
C*****************************************************************************00002200
   20 DO 21 L=1,MNMAXO                                                    00002210
      SMO=SMO+PHIX(L)**2                                                  00002220
     1     +2.0*PHX(L)*PHIX(L)                                            00002230
      SMT=SMT+PHIT(L)**2                                                  00002240
     1     +2.0*PHT(L)*PHIT(L)                                            00002250
   21 SMF=SMF+PHI(L)**2                                                   00002260
      SMO=SMO+PHIX(M)**2                                                  00002270
     1     +2.0*PHX(M)*PHIX(M)                                            00002280
      SMF=SMF+PHI(1)**2                                                   00002290
      SMT=SMT+PHIT(1)*(PHIT(1)+2.0*PHT(1))                                00002300
      SMR=SMR+(PHIX(1)*(PHIT(1)+PHT(1))+PHX(1)*PHIT(1))*2.0               00002310
      DO 122 L=2,MNMAXO,2                                                 00002320
  122 SMR=SMR+PHIX(L)*(PHIT(L+1)+PHT(L+1))+PHX(L)*PHIT(L+1)               00002330
      DO 121 L=3,MNMAXO,2                                                 00002340
  121 SMR=SMR+PHIX(L)*(PHIT(L-1)+PHT(L-1))+PHX(L)*PHIT(L-1)               00002350
C*****************************************************************************00002360
C     COMPILE BETA TERMS FOR THIS SPECIAL CASE                            *00002370
C*****************************************************************************00002380
      BX3(M)=SMO*.5                                                       00002390
      BT3(M)=SMT*.5                                                       00002400
      BE3(M)=SMF*.5                                                       00002410
      BXT3(1)=SMR*0.5                                                     00002420
      GO TO 9                                                             00002430
C*****************************************************************************00002440
C                                                                         *00002450
C     THIS SECTION HANDLES ASYMMETRIC MODES                               *00002460
C                                                                         *00002470
```

```
C*******************************************************************00002480
  101 MP=M+1                                                          00002490
      MAXL=MAXS(MP)                                                   00002500
C*******************************************************************00002510
C     TEST FOR PRESENCE OF SUMMATION COMBINATIONS                    *00002520
C*******************************************************************00002530
      IF (MAXL.EQ.0) GO TO 102                                        00002540
C*******************************************************************00002550
C     SET UP INDICES FOR SUMMATION COMBINATIONS                      *00002560
C*******************************************************************00002570
      DO 103 L=1,MAXL                                                 00002580
      I=IS(L,MP)                                                      00002590
      J=JS(L,MP)                                                      00002600
      II=I-1                                                          00002610
      JJ=J-1                                                          00002620
C*******************************************************************00002630
C     TEST FOR MODE 1 AND COMPILE SUMS                               *00002640
C*******************************************************************00002650
      IF (I.EQ.1) GO TO 103                                           00002660
      SMO=SMO+PHIX(I)*(PHIX(JJ)+PHX(JJ))+PHIX(J)*(PHIX(II)+           00002670
     1    PHX(II))+PHIX(II)*PHX(J)+PHIX(JJ)*PHX(I)                    00002680
      SMT=SMT+PHIT(I)*(PHIT(JJ)+PHT(JJ))+PHIT(J)*(PHIT(II)+           00002690
     1    PHT(II))+PHIT(II)*PHT(J)+PHIT(JJ)*PHT(I)                    00002700
      SMR=SMR+PHIX(I)*(PHIT(JJ)+PHT(JJ))+PHIX(J)*(PHIT(II)+           00002710
     1    PHT(II))-PHIX(II)*(PHIT(J)+PHT(J))-PHIX(JJ)*                00002720
     2    (PHIT(I)+PHT(I))+PHX(I)*PHIT(JJ)+PHX(J)*PHIT(II)-           00002730
     3    PHX(II)*PHIT(J)-PHX(JJ)*PHIT(I)                             00002740
      SMF=SMF+PHI(I)*PHI(JJ)+PHI(J)*PHI(II)                           00002750
  103 CONTINUE                                                        00002760
C*******************************************************************00002770
C     TEST FOR PRESENCE OF DIFFERENCE COMBINATIONS                   *00002780
C*******************************************************************00002790
  102 MAXL=MAXD(MP)                                                   00002800
      IF (MAXL.EQ.0) GO TO 104                                        00002810
C*******************************************************************00002820
C     SET UP INDICES FOR DIFFERENCE COMBINATIONS                     *00002830
C*******************************************************************00002840
      DO 105 L=1,MAXL                                                 00002850
      I=ID(L,MP)                                                      00002860
      J=JD(L,MP)                                                      00002870
      II=I-1                                                          00002880
      JJ=J-1                                                          00002890
C*******************************************************************00002900
C     TEST FOR MODE 1 AND COMPILE SUMS                               *00002910
C*******************************************************************00002920
      IF (J.EQ.1) GO TO 123                                           00002930
      SMO=SMO-PHIX(I)*(PHIX(JJ)+PHX(JJ))+PHIX(J)*(PHIX(II)+           00002940
     1    PHX(II))+PHIX(II)*PHX(J)-PHIX(JJ)*PHX(I)                    00002950
```

```
      SMT=SMT+PHIT(I)*(PHIT(JJ)+PHT(JJ))-PHIT(J)*(PHIT(II)+                00002960
     1     PHT(II))-PHIT(II)*PHT(J)+PHIT(JJ)*PHT(I)                        00002970
      SMR=SMR+PHIX(I)*(PHIT(JJ)+PHT(JJ))+PHIX(J)*(PHIT(II)+                00002980
     1     PHT(II))+PHIX(II)*(PHIT(J)+PHT(J))+PHIX(JJ)*                    00002990
     2     (PHIT(I)+PHT(I))+PHX(I)*PHIT(JJ)+PHX(J)*PHIT(II)+               00003000
     3     PHX(II)*PHIT(J)+PHX(JJ)*PHIT(I)                                 00003010
      SMF=SMF+PHI(I)*PHI(JJ)-PHI(J)*PHI(II)                                00003020
      GO TO 105                                                            00003030
C*************************************************************************00003040
C     EXECUTE BELOW IF J=1 IN DIFFERENCE COMBINATIONS                    *00003050
C*************************************************************************00003060
  123 SMO=SMO+(PHIX(1)*(PHIX(II)+PHX(II))+PHIX(II)*PHX(1))*2.0             00003070
      SMT=SMT+(PHIT(I)*(PHIT(1)+PHT(1))+PHIT(1)*PHT(II))*2.0               00003080
      SMR=SMR+(PHIX(1)*(PHIT(II)+PHT(II))+PHIX(II)*(PHIT(1)+               00003090
     1     PHT(1))+PHX(1)*PHIT(II)+PHX(II)*PHIT(1))*2.0                    00003100
      SMF=SMF+(PHI(I)*PHI(1))*2.0                                          00003110
  105 CONTINUE                                                             00003120
C*************************************************************************00003130
C     TEST FOR PRESENCE OF SAME-INDEX COMBINATION                        *00003140
C*************************************************************************00003150
  104 IF (MAXSY(MP).EQ.0) GO TO 10                                         00003160
C*************************************************************************00003170
C     SET UP INDICES AND COMPILE SUMS                                    *00003180
C*************************************************************************00003190
      I=IJS(MP)                                                            00003200
      II=I-1                                                               00003210
      SMO=SMO+PHIX(I)*(PHIX(II)+PHX(II))+PHIX(II)*PHX(I)                   00003220
      SMT=SMT+PHIT(I)*(PHIT(II)+PHT(II))+PHIT(II)*PHT(I)                   00003230
      SMR=SMR+PHIX(I)*(PHIT(II)+PHT(II))-PHIX(II)*(PHIT(I)+                00003240
     1     PHT(I))+PHX(I)*PHIT(II)-PHX(II)*PHIT(I)                         00003250
      SMF=SMF+PHI(I)*PHI(II)                                               00003260
C*************************************************************************00003270
C     COMPILE BETA TERMS                                                 *00003280
C*************************************************************************00003290
   10 BX3(M)=SMO                                                           00003300
      BT3(M)=SMT                                                           00003310
      BXT3(M)=SMR*.5                                                       00003320
      BE3(M)=SMF                                                           00003330
    9 CONTINUE                                                             00003340
 2222 CONTINUE                                                           00003350
      RETURN                                                               00003360
      END                                                                  00003370
```

```
      SUBROUTINE MODES                                                  00000010
C*****************************************************************    00000020
C          IN THIS SUBROUTINE, ARRAYS THAT DEFINE THOSE SETS OF INDICES  *00000030
C          THAT COMBINE TO EQUAL EACH VALUE OF N IN THE PROBLEM ARE DETER-00000040
C          MINED.  MODES IS CALLED PRIOR TO THE FIRST ITERATION AND AFTER*00000050
C          EVERY ITERATION UNTIL A SPECIFIED NUMBER OF FOURIER TERMS IS   *00000060
C          REACHED.  EACH FOURIER INDEX IN THE PROBLEM IS SUBTRACTED FROM*00000070
C          ALL OTHER FOURIER INDICES AND THE RESULT IS COMPARED WITH ALL *00000080
C          FOURIER INDICES TO SEE IF THE NEW VALUE EXISTS IN THE PROGRAM.*00000090
C          IF IT DOES, THE LOCATIONS OF THE TWO INDICES THAT MADE THE COM-00000100
C          BINATION ARE STORED IN TWO SPECIAL 2-DIMENSIONAL ARRAYS, ID ANDO0000110
C          JD.  ONE ARGUMENT OF EACH ARRAY IS THE VALUE OF THE NEW INDEX *00000120
C          AND THE OTHER IS THE NUMBER OF COMBINATIONS OF INDICES THAT    *00000130
C          ALSO GIVE THIS VALUE OF THE INDEX. IF THERE IS NO INDEX IN THE*00000140
C          PROGRAM THAT MATCHES THE NEW ONE, THEN A NEW FOURIER TERM HAS *00000150
C          BEEN GENERATED AND WILL BE CONSIDERED IN THE NEXT ITERATION FORO0000160
C          SOLUTION. THE VARIABLE MAXD STORES THE TOTAL NUMBER OF SUCH    *00000170
C          COMBINATIONS FOR EACH VALUE OF THE FOURIER INDEX.              *00000180
C               IN A SIMILAR MANNER, EACH INDEX IS ADDED TO EVERY OTHER   *00000190
C          INDEX AND THE SUM COMPARED WITH ALL INDICES.  THIS RESULT IS   *00000200
C          STORED IN THE 2-DIMENSIONAL ARRAYS 'IS' AND 'JS', IN THE SAME *00000210
C          MANNER AS WAS DONE FOR THE SUBTRACTION CASE.  THE VARIABLE     *00000220
C          'MAXS' STORES THE TOTAL NUMBER OF SUMMATION COMBINATIONS FOR   *00000230
C          EACH VALUE OF THE FOURIER INDEX.                               *00000240
C               A SPECIAL ROUTINE HANDLES THE CASE WHERE THE INDEX IS ADD-00000250
C          ED TO ITSELF, THE SUM COMPARED WITH ALL OTHER INDICES.  THE 1-*00000260
C          DIMENSIONAL ARRAY 'IJS' STORES THE LOCATION OF THE INDEX AND   *00000270
C          THE VARIABLE 'MAXSY' INDICATES IF THIS COMBINATION EXISTS.     *00000280
C          SINCE THE INDEX, SUBTRACTED FROM ITSELF, ALWAYS GIVES MODE '0',00000290
C          NO SPECIFIC STATEMENTS ARE REQUIRED IN THAT CASE.              *00000300
C               WITH THIS PROCEDURE, THE SERIES OF PRODUCTS THAT MAKE UP  *00000310
C          THE BETA'S AND AETA'S CONTAIN NO ZERO TERM, AND THE SUMMATION  *00000320
C          IS CARRIED OUT IN PHIBET(K) AND TEAETA(K) OVER SPECIFICALLY    *00000330
C          DEFINDED LIMITS.                                               *00000340
C               THE VARIABLE 'JUMP' IS '1' UNLESS SPECIFICALLY SET TO '2'*00000350
C          ON INPUT CARDS, INDICATING PROCESSING OF GENERAL ASYMMETRIC    *00000360
C          LOADS AND IMPERFECTIONS.  IN SATANS-II, THE NUMBER OF MODES IS*00000370
C          LIMITED TO 25.                                                 *00000380
C*****************************************************************    00000390
      COMMON /IBL1/ MNMAX                                               00000400
      COMMON /IBL2/ N(25),MNINIT                                        00000410
```

```
      COMMON /IBL7/ MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),        00000420
     1              JS(25,25),ID(25,25),JD(25,25),IJS(25)                00000430
      COMMON /IBL9/ MAXM                                                  00000440
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                             00000450
C*************************************************************************00000460
C           DETERMINE IF NEED FOR MODAL COUPLING EXISTS                  *00000470
C*************************************************************************00000480
      IF(MAXM.EQ.1) RETURN                                                00000490
C*************************************************************************00000500
C           SEE IF ENOUGH NEW MODES HAVE ALREADY BEEN GENERATED          *00000510
C*************************************************************************00000520
      IF(MNINIT.GT.MAXM) RETURN                                           00000530
C*************************************************************************00000540
C                                                                        *00000550
C                                                                        *00000560
C           THIS IS SECTION HANDLES MODES GENERATED FROM THE DIFFERENCE  *00000570
C           COMBINATIONS OF EXISTING MODES                               *00000580
C                                                                        *00000590
C                                                                        *00000600
C*************************************************************************00000610
C           SET UP INITIAL LOOP FOR DIFFERENCE COMBINATIONS              *00000620
C*************************************************************************00000630
      DO 1 MN=1,MNMAXO,JUMP                                               00000640
      NMN=N(MN)                                                           00000650
      NNS=MN                                                              00000660
      IF(MNINIT.GT.MN) NNS=MNINIT                                         00000670
C*************************************************************************00000680
C           SET UP SECOND LOOP - TO SUBTRACT PRESENT MODE FROM ALL OTHER *00000690
C           MODES                                                        *00000700
C*************************************************************************00000710
      DO 1 MM=NNS,MNMAXO,JUMP                                             00000720
      NMM=N(MM)                                                           00000730
C*************************************************************************00000740
C           CREATE DIFFERENCE                                            *00000750
C*************************************************************************00000760
      NTEST=IABS(NMN-NMM)                                                 00000770
C*************************************************************************00000780
C           SET UP 3RD LOOP - TO COMPARE DIFFERENCE WITH ALL OTHER MODES *00000790
C*************************************************************************00000800
      DO 2 MMFT=1,MNMAX,JUMP                                              00000810
C*************************************************************************00000820
C           IF WE SATISFY HERE, MODE EXISTS, GO TO INCREMENT 'MAXD'      *00000830
C*************************************************************************00000840
      IF(NTEST.EQ.N(MMFT)) GO TO 10                                       00000850
    2 CONTINUE                                                            00000860
C*************************************************************************00000870
C           IF WE MAKE IT TO HERE, WE HAVE GENERATED A NEW MODE          *00000880
C*************************************************************************00000890
```

```
C           DO WE WANT ANY MORE NEW MODES                                    *00000900
C*****************************************************************************00000910
      IF(ICORFL.EQ.1) GO TO 1                                                  00000920
C*****************************************************************************00000930
C           FROM HERE TO '1', WE INCREMENT COUNTERS AND MAKE APPROPRIATE*00000940
C           ENTRIES IN 'MAXD','ID' & 'JD'.  TO SIMPLIFY USAGE OF 'ID'    *00000950
C           AND 'JD' IN TEAETA AND PHIBET, MODES ARE PLACED IN THESE     *00000960
C           ARRAYS SUCH THAT THE HIGHER MODE NUMBER IS ALWAYS FIRST      *00000970
C*****************************************************************************00000980
      MNMAX=MNMAX+JUMP                                                         00000990
      N(MNMAX)=NTEST                                                           00001000
      IF (JUMP.GT.1) N(MNMAX-1)=-NTEST                                         00001010
      MMFT=MNMAX                                                               00001020
      IF(MNMAX.EQ.MAXM) ICORFL=1                                               00001030
   10 IF(NMN-NMM) 11,1,12                                                      00001040
   11 LOCD=MAXD(MMFT)+1                                                        00001050
      MAXD(MMFT)=LOCD                                                          00001060
      ID(LOCD,MMFT)=MM                                                         00001070
      JD(LOCD,MMFT)=MN                                                         00001080
      GO TO 1                                                                  00001090
   12 LOCD=MAXD(MMFT)+1                                                        00001100
      MAXD(MMFT)=LOCD                                                          00001110
      ID(LOCD,MMFT)=MN                                                         00001120
      JD(LOCD,MMFT)=MM                                                         00001130
    1 CONTINUE                                                                 00001140
C*****************************************************************************00001150
C                                                                              *00001160
C                                                                              *00001170
C           THIS SECTION HANDLES MODES GENERATED FROM THE SUMMATION          *00001180
C           COMBINATIONS OF EXISTING MODES                                   *00001190
C                                                                              *00001200
C                                                                              *00001210
C*****************************************************************************00001220
C           NOW SET UP INITIAL LOOP FOR SUM COMBINATIONS                     *00001230
C*****************************************************************************00001240
      DO 301 MN=1,MNMAXO,JUMP                                                  00001250
      NMN=N(MN)                                                                00001260
      NNS=MN                                                                   00001270
      IF(MNINIT.GT.MN) NNS=MNINIT                                              00001280
C*****************************************************************************00001290
C           SET UP SECOND LOOP - TO ADD PRESENT MODE TO ALL OTHER MODES *00001300
C*****************************************************************************00001310
      DO 301 MM=NNS,MNMAXO,JUMP                                                00001320
      NMM=N(MM)                                                                00001330
C*****************************************************************************00001340
C           CREATE SUM                                                       *00001350
C*****************************************************************************00001360
      NTEST=NMN+NMM                                                            00001370
```

```
C*****************************************************************00001380
C          SET UP THIRD LOOP - TO COMPARE SUM WITH ALL OTHER MODES     *00001390
C*****************************************************************00001400
      DO 302 MMFT=1,MNMAX,JUMP                                        00001410
C*****************************************************************00001420
C          IF WE SATISFY HERE, MODE EXISTS, GO TO INCREMENT 'MAXS' (OR *00001430
C          'MAXSY')                                                 *00001440
C*****************************************************************00001450
      IF(NTEST.EQ.N(MMFT)) GO TO 310                                  00001460
  302 CONTINUE                                                        00001470
C*****************************************************************00001480
C          IF WE MAKE IT TO HERE, WE HAVE GENERATED A NEW MODE        *00001490
C*****************************************************************00001500
C          DO WE WANT ANY MORE NEW MODES                            *00001510
C*****************************************************************00001520
      IF(ICORFL.EQ.1) GO TO 301                                       00001530
      IF(MNMAX.GE.MAXM) GO TO 301                                     00001540
C*****************************************************************00001550
C          INCREMENT 'MNMAX' AND ESTABLISH NEW MODE NUMBER          *00001560
C*****************************************************************00001570
      MNMAX=MNMAX+JUMP                                                00001580
      N(MNMAX)=NTEST                                                  00001590
      IF (JUMP.GT.1) N(MNMAX-1)=-NTEST                                00001600
      MMFT=MNMAX                                                      00001610
      IF(MNMAX.GE.MAXM) ICORFL=1                                      00001620
C*****************************************************************00001630
C          IF MODE WAS ADDED TO ITSELF, GO TO 'MAXSY & IJS' SECTION   *00001640
C*****************************************************************00001650
  310 IF(NMN.EQ.NMM) GO TO 360                                        00001660
C*****************************************************************00001670
C          MAKE ENTRIES IN 'LOCS','IS' & 'JS'                       *00001680
C*****************************************************************00001690
      LOCS=MAXS(MMFT)+1                                               00001700
      MAXS(MMFT)=LOCS                                                 00001710
      IS(LOCS,MMFT)=MN                                                00001720
      JS(LOCS,MMFT)=MM                                                00001730
      GO TO 301                                                       00001740
C*****************************************************************00001750
C          SEE IF THE SUM OF THE MODE WITH ITSELF WAS THE 0-TH MODE   *00001760
C*****************************************************************00001770
  360 IF(NMN.EQ.0) GO TO 301                                          00001780
C*****************************************************************00001790
C          IF HERE, IT WASN'T, MAKE ENTRIES IN 'MAXSY' AND 'IJS'      *00001800
C*****************************************************************00001810
      MAXSY(MMFT)=1                                                   00001820
      IJS(MMFT)=MN                                                    00001830
  301 CONTINUE                                                        00001840
      MNINIT=MNMAXO+JUMP                                              00001850
```

```
      IF(ICORFL.GT.0) IPASS=IPASS+1                                     00001860
      IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRX                    00001870
      RETURN                                                            00001880
      END                                                               00001890
```

```
      SUBROUTINE INLPOL                                                 00000010
C*************************************************************************00000020
C       THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,       *00000030
C     -SUB THETA, -SUB S-THETA, AETA-SUB S-S AND -SUB THETA-S AT AN     *00000040
C       INITIAL POLE.                                                   *00000050
C*************************************************************************00000060
      COMMON /IBL1/ MNMAX                                               00000070
      COMMON /IBL3/ MO,M1,M2,M3                                         00000080
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                           00000090
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000100
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000110
      COMMON /BL5/  TT(25),EMT(25),DT(25),DMT(25)                       00000120
      COMMON /BL6/  Z(4,1001),SOE,OSE,ALOAD                             00000130
      COMMON /BL7/  D1,S1                                               00000140
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                00000150
      COMMON /BL11A/ PHEN,T2N                                           00000160
      COMMON /BL17/ DEL                                                 00000170
      COMMON /BL29/ BX1(25),BT1(25),BXT1(25),BE1(25),BX2(25),BT2(25),   00000180
     1              BXT2(25),BE2(25)                                    00000190
      COMMON /BL30/ EXX1(25),ETT1(25),ETX1(25),EX1(25),ET1(25),EXX2(25),00000200
     1              ETT2(25),ETX2(25),EXT2(25),EX2(25),ET2(25)          00000210
      COMMON /BL31/ DELSQ,EXT1(25)                                      00000220
C*************************************************************************00000230
      IF (JUMP.EQ.2) GO TO 1000                                         00000240
      DO 1 MN=1,MNMAX                                                   00000250
      BX1 (MN)=0.                                                       00000260
      BT1 (MN)=0.                                                       00000270
      BXT1(MN)=0.                                                       00000280
      BE1 (MN)=0.                                                       00000290
      EX1 (MN)=0.                                                       00000300
      ET1 (MN)=0.                                                       00000310
      ETX1(MN)=0.                                                       00000320
```

```
    1 EXX1(MN)=0.                                                                  00000330
      IF(M1.EQ.0) RETURN                                                           00000340
      I2=2+(M1-1)*KMAX2                                                            00000350
      I3=I2+1                                                                      00000360
      I4=I3+1                                                                      00000370
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z(1,I2)                 00000380
      BET=.5*PHEE**2                                                               00000390
      IF(ITRMAX.EQ.1) BET=0.                                                       00000400
      T2=0.                                                                        00000410
      IF(M2.EQ.0) GO TO 2                                                          00000420
      CALL BDB(1,B,DB,D,DD)                                                        00000430
      I2=2+(M2-1)*KMAX2                                                            00000440
      I3=I2+1                                                                      00000450
      I4=I3+1                                                                      00000460
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BET)                00000470
      Q1=.5*PHEE*T2                                                                00000480
      BX1(M2)=BET                                                                  00000490
      BT1(M2)=-BET                                                                 00000500
      BXT1(M2)=-BET                                                                00000510
      ETX1(M1)=Q1                                                                  00000520
      IF(M3.EQ.0) GO TO 2                                                          00000530
      EXX1(M3)=Q1                                                                  00000540
      ETX1(M3)=-Q1                                                                 00000550
    2 TO=0.                                                                        00000560
      IF(M0.EQ.0) GO TO 3                                                          00000570
      BX1(M0)=BET                                                                  00000580
      BT1(M0)=BET                                                                  00000590
      CALL BDB(1,B,DB,D,DD)                                                        00000600
      CALL TLOAD(1)                                                                00000610
      I2=2+(M0-1)*KMAX2                                                            00000620
      I3=I2+1                                                                      00000630
      I4=I3+1                                                                      00000640
      TO=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I2)            00000650
     1+.5*SOE*BET)-TT(M0)*ALOAD                                                    00000660
    3 EXX1(M1)=PHEE*(TO+.5*T2)                                                     00000670
      RETURN                                                                       00000680
 1000 DO 1001 MN=1,MNMAX                                                           00000690
      BX1 (MN)=0.                                                                  00000700
      BT1 (MN)=0.                                                                  00000710
      BXT1(MN)=0.                                                                  00000720
      BE1 (MN)=0.                                                                  00000730
      EX1 (MN)=0.                                                                  00000740
      ET1 (MN)=0.                                                                  00000750
      ETX1(MN)=0.                                                                  00000760
 1001 EXX1(MN)=0.                                                                  00000770
      IF(M1.EQ.0) RETURN                                                           00000780
      I2=2+(M1-1)*KMAX2                                                            00000790
      I3=I2+1                                                                      00000800
```

```
      I4=I3+1                                                                   00000810
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z(1,I2)              00000820
      T2=0.                                                                     00000830
      IF(M2.EQ.0) GO TO 1002                                                    00000840
      CALL BDB(1,B,DB,D,DD)                                                     00000850
      I2=2+(M2-1)*KMAX2                                                         00000860
      I3=I2+1                                                                   00000870
      I4=I3+1                                                                   00000880
      PHX1=PHIXB(KMAX+1)                                                        00000890
      PHX2=PHIXB(2*KMAX+1)                                                      00000900
      PHEN=(1.5*Z(3,I2-KMAX2)-2.*Z(3,I3-KMAX2)+.5*Z(3,I4-KMAX2))/DEL+           00000910
     1OMXI(1)*Z(1,I2-KMAX2)                                                     00000920
      BX1(M2)=.5*(PHEE*(PHEE+2.*PHX1)-PHEN*(PHEN+2.*PHX2))                      00000930
      IF(ITRMAX.EQ.1) BX1(M2)=0.                                                00000940
      BT1(M2)=-BX1(M2)                                                          00000950
      BXT1(M2)=-BX1(M2)                                                         00000960
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BX1(M2))         00000970
      M2L=M2-1                                                                  00000980
      BX1(M2L)=PHEE*(PHEN+PHX2)+PHX1*PHEN                                       00000990
      IF(ITRMAX.EQ.1) BX1(M2L)=0.                                               00001000
      BT1(M2L)=-BX1(M2L)                                                        00001010
      BXT1(M2L)=BX1(M2L)                                                        00001020
      T2N=B*D1*((-1.5*Z(1,I2-KMAX2)+2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))          00001030
     1  /DEL+.5*SOE*BX1(M2L))                                                   00001040
 1002 TO=0.                                                                     00001050
      IF(MO.EQ.0) GO TO 1003                                                    00001060
      BX1(MO)=.5*(PHEE*(PHEE+2.*PHX1)+PHEN*(PHEN+2.*PHX2))                      00001070
      IF(ITRMAX.EQ.1) BX1(MO)=0.                                                00001080
      BT1(MO)=BX1(MO)                                                           00001090
      CALL BDB(1,B,DB,D,DD)                                                     00001100
      CALL TLOAD(1)                                                             00001110
      I2=2+(MO-1)*KMAX2                                                         00001120
      I3=I2+1                                                                   00001130
      I4=I3+1                                                                   00001140
      TO=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I2)         00001150
     1+.5*SOE*BX1(MO))-TT(MO)*ALOAD                                             00001160
 1003 IF(ITRMAX.EQ.1) RETURN                                                    00001170
      PHSS=PHEE+PHX1                                                            00001180
      PHSP=PHEN+PHX2                                                            00001190
      M1L=M1-1                                                                  00001200
      EXX1(M1)=PHSS*TO+.5*(PHSS*T2+PHSP*T2N)                                    00001210
      EXX1(M1L)=PHSP*TO-.5*(PHSP*T2-PHSS*T2N)                                   00001220
      ETX1(M1)=.5*(PHSS*T2+PHSP*T2N)                                            00001230
      ETX1(M1L)=.5*(-PHSP*T2+PHSS*T2N)                                          00001240
      IF(M3.EQ.0) RETURN                                                        00001250
      M3L=M3-1                                                                  00001260
      EXX1(M3)=.5*(PHSS*T2-PHSP*T2N)                                            00001270
      EXX1(M3L)=.5*(PHSS*T2N+PHSP*T2)                                           00001280
```

```
      ETX1(M3)=.5*(-PHSS*T2+PHSP*T2)                                    00001290
      ETX1(M3L)=.5*(-PHSP*T2-PHSS*T2N)                                  00001300
      RETURN                                                            00001310
      END                                                               00001320
```

```
      SUBROUTINE FNLPOL                                                 00000010
C*****************************************************************    00000020
C       THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,      *00000030
C       -SUB THETA, -SUB S-THETA, AETA-SUB S-S, AND -SUB THETA-S AT A  00000040
C       FINAL POLE.                                                   *00000050
C*****************************************************************    00000060
      COMMON /IBL1/ MNMAX                                               00000070
      COMMON /IBL3/ M0,M1,M2,M3                                         00000080
      COMMON /IBL4/ KMAX,KL                                             00000090
      COMMON /IBL11/ ICOREL,IPASS,JUMP,MPERFS                           00000100
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000110
      COMMON /IBL13/ ITRMAX,LSMAX                                       00000120
      COMMON /BL5/  TT(25),EMT(25),DT(25),DMT(25)                       00000130
      COMMON /BL6/  Z(4,1001),SOE,OSE,ALOAD                             00000140
      COMMON /BL7/  D1,S1                                               00000150
      COMMON /BL11A/ PHEN,T2N                                           00000160
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                00000170
      COMMON /BL17/ DEL                                                 00000180
      COMMON /BL27/ BX3(25),BT3(25),BXT3(25),BE3(25)                    00000190
      COMMON /BL28/ EXX3(25),ETT3(25),ETX3(25),EXT3(25),EX3(25),ET3(25) 00000200
C*****************************************************************    00000210
      IF (JUMP.EQ.2) GO TO 1000                                         00000220
      DO 1 MN=1,MNMAX                                                   00000230
      BX3 (MN)=0.                                                       00000240
      BT3 (MN)=0.                                                       00000250
      BXT3(MN)=0.                                                       00000260
      BE3 (MN)=0.                                                       00000270
      EX3 (MN)=0.                                                       00000280
      ET3 (MN)=0.                                                       00000290
      ETX3(MN)=0.                                                       00000300
    1 EXX3(MN)=0.                                                       00000310
      CALL BDB(KMAX,B,DB,D,DB)                                          00000320
```

```
      IF(M1.EQ.0) RETURN                                                        00000330
      KM=KMAX1+(M1-1)*KMAX2                                                     00000340
      KM1=KM-1                                                                  00000350
      KM2=KM-2                                                                  00000360
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(KMAX)*Z(1,KM)        00000370
      BET=.5*PHEE**2                                                            00000380
      IF(ITRMAX.EQ.1) BET=0.                                                    00000390
      T2=0.                                                                     00000400
      IF(M2.EQ.0) GO TO 2                                                       00000410
      KM=KMAX1+(M2-1)*KMAX2                                                     00000420
      KM1=KM-1                                                                  00000430
      KM2=KM-2                                                                  00000440
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SOE*BET)            00000450
      Q1=.5*PHEE*T2                                                             00000460
      BX3(M2)=BET                                                               00000470
      BT3(M2)=-BET                                                              00000480
      BXT3(M2)= BET                                                             00000490
      ETX3(M1)=Q1                                                               00000500
      IF(M3.EQ.0) GO TO 2                                                       00000510
      EXX3(M3)=Q1                                                               00000520
      ETX3(M3)=-Q1                                                              00000530
    2 TO=0.                                                                     00000540
      IF(MO.EQ.0) GO TO 3                                                       00000550
      CALL TLOAD(KMAX)                                                          00000560
      KM=KMAX1+(MO-1)*KMAX2                                                     00000570
      KM1=KM-1                                                                  00000580
      KM2=KM-2                                                                  00000590
      BX3(MO)=BET                                                               00000600
      BT3(MO)=BET                                                               00000610
      TO=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+OMXI(KMAX)*            00000620
     1Z(3,KM)+.5*SOE*BET)-TT(MO)*ALOAD                                          00000630
    3 EXX3(M1)=PHEE*(TO+.5*T2)                                                  00000640
      RETURN                                                                    00000650
 1000 DO 1001 MN=1,MNMAX                                                        00000660
      BX3 (MN)=0.                                                               00000670
      BT3 (MN)=0.                                                               00000680
      BXT3(MN)=0.                                                               00000690
      BE3 (MN)=0.                                                               00000700
      EX3 (MN)=0.                                                               00000710
      ET3 (MN)=0.                                                               00000720
      ETX3(MN)=0.                                                               00000730
 1001 EXX3(MN)=0.                                                               00000740
      CALL BDB(KMAX,B,DB,D,DB)                                                  00000750
      IF(M1.EQ.0) RETURN                                                        00000760
      KM=KMAX1+(M1-1)*KMAX2                                                     00000770
      KM1=KM-1                                                                  00000780
      KM2=KM-2                                                                  00000790
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(KMAX)*Z(1,KM)        00000800
```

```
      T2=0.                                                                    00000810
      IF(M2.EQ.0) GO TO 1002                                                   00000820
      KM=KMAX1+(M2-1)*KMAX2                                                    00000830
      KM1=KM-1                                                                 00000840
      KM2=KM-2                                                                 00000850
      J=KMAX*2                                                                 00000860
      I=J+KMAX                                                                 00000870
      PHX1=PHIXB(J)                                                            00000880
      PHX2=PHIXB(I)                                                            00000890
      PHEN=-(1.5*Z(3,KM-KMAX2)-2.*Z(3,KM1-KMAX2)+.5*Z(3,KM2-KMAX2))/DEL       00000900
     1+OMXI(KMAX)*Z(1,KM-KMAX2)                                                00000910
      BX3(M2)=.5*(PHEE*(PHEE+2.*PHX1)-PHEN*(PHEN+2.*PHX2))                     00000920
      IF(ITRMAX.EQ.1) BX3(M2)=0.                                               00000930
      BT3(M2)=-BX3(M2)                                                         00000940
      BXT3(M2)=BX3(M2)                                                         00000950
      M2L=M2-1                                                                 00000960
      BX3(M2L)=PHEE*(PHEN+PHX2)+PHX1*PHEN                                      00000970
      IF(ITRMAX.EQ.1) BX3(M2L)=0.                                              00000980
      BT3(M2L)=-BX3(M2L)                                                       00000990
      BXT3(M2L)=-BX3(M2L)                                                      00001000
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SOE*BX3(M2))      00001010
      T2N=B*D1*((1.5*Z(1,KM-KMAX2)-2.*Z(1,KM1-KMAX2)+.5*Z(1,KM2-KMAX2))       00001020
     1   /DEL+.5*SOE*BX3(M2L))                                                 00001030
1002  T0=0.                                                                    00001040
      IF(M0.EQ.0) GO TO 1003                                                   00001050
      CALL TLOAD(KMAX)                                                         00001060
      KM=KMAX1+(M0-1)*KMAX2                                                    00001070
      KM1=KM-1                                                                 00001080
      KM2=KM-2                                                                 00001090
      BX3(M0)=.5*(PHEE*(PHEE+2.*PHX1)+PHEN*(PHEN+2.*PHX2))                     00001100
      IF(ITRMAX.EQ.1) BX3(M0)=0.                                               00001110
      BT3(M0)=BX3(M0)                                                          00001120
      T0=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+OMXI(KMAX)*           00001130
     1 Z(3,KM)+.5*SOE*BX3(M0))-TT(M0)*ALOAD                                    00001140
1003  IF(ITRMAX.EQ.1) RETURN                                                   00001150
      PHSS=PHEE+PHX1                                                           00001160
      PHSP=PHEN+PHX2                                                           00001170
      M1L=M1-1                                                                 00001180
      EXX3(M1)=PHSS*T0+.5*(PHSS*T2+PHSP*T2N)                                   00001190
      EXX3(M1L)=PHSP*T0-.5*(PHSP*T2-PHSS*T2N)                                  00001200
      ETX3(M1)=.5*(PHSS*T2+PHSP*T2N)                                           00001210
      ETX3(M1L)=.5*(-PHSP*T2+PHSS*T2N)                                         00001220
      IF(M3.EQ.0) RETURN                                                       00001230
      M3L=M3-1                                                                 00001240
      EXX3(M3)=.5*(PHSS*T2-PHSP*T2N)                                           00001250
      EXX3(M3L)=.5*(PHSS*T2N+PHSP*T2)                                          00001260
      ETX3(M3)=.5*(-PHSS*T2+PHSP*T2N)                                          00001270
      ETX3(M3L)=.5*(-PHSP*T2-PHSS*T2N)                                         00001280
```

```
      RETURN                                                            00001290
      END                                                               00001300


      SUBROUTINE POLE(K)                                                00000010
C*************************************************************************00000020
C        THIS SUBROUTINE PRINTS THE SOLUTION AT AN INITIAL AND A FINAL *00000030
C        POLE.                                                         *00000040
C*************************************************************************00000050
      IMPLICIT LOGICAL*1 ($)                                            00000060
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                 00000070
      COMMON /IBL2/ N(25),MNINIT                                        00000080
      COMMON /IBL3/ M0,M1,M2,M3                                         00000090
      COMMON /IBL4/ KMAX,KL                                             00000100
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000110
      COMMON /IBL7/ MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),       00000120
     1              JS(25,25),ID(25,25),JD(25,25),IJS(25)               00000130
      COMMON /IBL8/ LSTEP,ITR                                           00000140
      COMMON /IBL10/ IFREQ,NTHMAX                                       00000150
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                           00000160
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00000170
      COMMON /BL4/ P(4,4,951),X(4,951),ZF1M(4,4,25),ZF2M(4,4,25),       00000180
     1             ZF3M(4,4,25),ZF4M(4,4,25)                          00000190
      COMMON /BL5/  TT(25),MT(25),DT(25),DMT(25)                        00000200
      COMMON /BL6/  Z(4,100),SOE,OSE,ALOAD                              00000210
      COMMON /BL7/  D1,S1                                               00000220
      COMMON /BL8/  R(100),GAM(100),OMT(100)                            00000230
      COMMON /BL10/ PHIX(25),PHIT(25),PHI(25)                           00000240
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                00000250
      COMMON /BL11A/ PHEN,T2N                                           00000260
      COMMON /BL12/ TDLI,TDEL                                           00000270
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    00000280
      COMMON /BL15/ NU,U1(25),V1(25),W1(25),V2(25),U2(25),W2(25),U3(25),00000290
     1              V3(25),W3(25)                                       00000300
      COMMON /BL17/ DEL                                                 00000310
      COMMON /BL19/ TH(36)                                              00000320
      COMMON /BL20/ DEOMX(100)                                          00000330
      COMMON /BL27/ BX3(25),BT3(25),BXT3(25),BE3(25)                    00000340
```

```
      COMMON /BL31/ DELSQ,EXT1(25)                                          00000350
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                     00000360
      COMMON /BL100/ TEEO,$DYNMC                                            00000370
      COMMON /BL101/ ZO(4,1001),Z2(4,1001),Z3(4,1001),DELSD                 00000380
      COMMON /BL102/ DELOAD                                                 00000390
      COMMON /BL103/ MASS(100)                                              00000400
      COMMON /BL110/ TX(25),TTH(25),TXT(25),MX(25),MTH(25),MXT(25),         00000410
     1               QS(25)                                                 00000420
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                                 00000430
C*********************************************************************    00000440
      IF (JUMP.EQ.2) GO TO 1000                                             00000450
      CALL BDB(K,BS,DB,DS,DD)                                               00000460
      IF(K.EQ.KMAX) GO TO 301                                               00000470
      DO 202 MN=1,MNMAXO                                                    00000480
      U1(MN)=U2(MN)                                                         00000490
      V1(MN)=V2(MN)                                                         00000500
      W1(MN)=W2(MN)                                                         00000510
      I3=3+(MN-1)*KMAX2                                                     00000520
      I2=I3-1                                                               00000530
      U2(MN)=Z(1,I3)                                                        00000540
      V2(MN)=Z(2,I3)                                                        00000550
      W2(MN)=Z(3,I3)                                                        00000560
      PHIX(MN)=0.                                                           00000570
      PHIT(MN)=0.0                                                          00000580
      PHI(MN)=0.                                                            00000590
      MX(MN)=Z(4,I2)*ABZ3                                                   00000600
      MTH(MN)=0.                                                            00000610
      MXT(MN)=0.                                                            00000620
      QS(MN)=0.                                                             00000630
      TX(MN)=0.                                                             00000640
      TTH(MN)=0.                                                            00000650
  202 TXT(MN)=0.                                                            00000660
      IF(M1.EQ.0) GO TO 203                                                 00000670
      CALL INLPOL                                                           00000680
      PHIX(M1)=PHEE*ABZO                                                    00000690
      PHIT(M1)=-PHEE*ABZO                                                   00000700
      QS(M1)=0.                                                             00000710
      IF(MO.EQ.0) GO TO 204                                                 00000720
       TX(MO) =TO*ABZ                                                       00000730
      TTH(MO) =TO*ABZ                                                       00000740
      MTH(MO) =MX(MO)                                                       00000750
  204 IF(M2.EQ.0) GO TO 205                                                 00000760
       TX(M2) = T2*ABZ                                                      00000770
      TTH(M2) =-T2*ABZ                                                      00000780
      TXT(M2)=-T2*ABZ                                                       00000790
      MTH(M2) =-MX(M2)                                                      00000800
      MXT(M2) =MTH(M2)                                                      00000810
      GO TO 205                                                             00000820
```

```
203 IF(M0.EQ.0) GO TO 206                                                    00000830
    I3=3+(M0-1)*KMAX2                                                        00000840
    I4=I3+1                                                                  00000850
    CALL TLOAD(1)                                                            00000860
    TX(M0)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ         00000870
   1       -TT(M0)*ABZ*ALOAD                                                 00000880
    TTH(M0)=TX(M0)                                                           00000890
    MTH(M0)=MX(M0)                                                           00000900
206 IF(M2.EQ.0) GO TO 205                                                    00000910
    I3=3+(M2-1)*KMAX2                                                        00000920
    I4=I3+1                                                                  00000930
    TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL                                 00000940
     TX(M2) = TX(M2)*ABZ                                                     00000950
    TTH(M2) =-TX(M2)                                                         00000960
    TXT(M2) =-TX(M2)                                                         00000970
    MTH(M2) =-MX(M2)                                                         00000980
    MXT(M2) =-MX(M2)                                                         00000990
205 RETURN                                                                   00001000
301 CONTINUE                                                                 00001010
    DO 302 MN=1,MNMAXO                                                       00001020
    U1(MN)=U2(MN)                                                            00001030
    V1(MN)=V2(MN)                                                            00001040
    W1(MN)=W2(MN)                                                            00001050
    PHIX(MN)=0.                                                              00001060
    PHIT(MN)=0.                                                              00001070
    PHI(MN)=0.                                                               00001080
    IK=KMAX1+(MN-1)*KMAX2                                                    00001090
    MX(MN)=Z(4,IK)*ABZ3                                                      00001100
    MTH(MN)=0.                                                               00001110
    MXT(MN)=0.                                                               00001120
    QS(MN)=0.                                                                00001130
    TX(MN)=0.                                                                00001140
    TTH(MN)=0.                                                               00001150
302 TXT(MN)=0.                                                               00001160
    IF(M1.EQ.0) GO TO 303                                                    00001170
    CALL FNLPOL                                                              00001180
    PHIX(M1)=PHEE*ABZO                                                       00001190
    PHIT(M1)= PHEE*ABZO                                                      00001200
    QS(M1)=0.                                                                00001210
    IF(M0.EQ.0) GO TO 304                                                    00001220
     TX(M0) =T0*ABZ                                                          00001230
    TTH(M0) =T0*ABZ                                                          00001240
    MTH(M0) =MX(M0)                                                          00001250
304 IF(M2.EQ.0) GO TO 305                                                    00001260
     TX(M2) = T2*ABZ                                                         00001270
    TTH(M2) =-T2*ABZ                                                         00001280
    TXT(M2) = T2*ABZ                                                         00001290
    MTH(M2) =-MX(M2)                                                         00001300
```

```
      MXT(M2) =MX(M2)                                                          00001310
      GO TO 305                                                                00001320
  303 IF(MO.EQ.0) GO TO 306                                                    00001330
      IKM=KMAX+(MO-1)*KMAX2                                                    00001340
      IM1=IKM-1                                                                00001350
      CALL TLOAD(KMAX)                                                         00001360
      TX(MO)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+OMXI(KMAX)*Z(3,IKM+1)       00001370
     1        )*ABZ-TT(MO)*ABZ*ALOAD                                           00001380
      TTH(MO) =TX(MO)                                                          00001390
      MTH(MO) =MX(MO)                                                          00001400
  306 IF(M2.EQ.0) GO TO 305                                                    00001410
      IKM=KMAX+(M2-1)*KMAX2                                                    00001420
      IM1=IKM-1                                                                00001430
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL                              00001440
      TX(M2)=TX(M2)*ABZ                                                        00001450
      TTH(M2) =-TX(M2)                                                         00001460
      TXT(M2) = TX(M2)                                                         00001470
      MTH(M2) =-MX(M2)                                                         00001480
      MXT(M2) = MX(M2)                                                         00001490
  305 RETURN                                                                   00001500
 1000 CALL BDB(K,BS,DB,DS,DD)                                                  00001510
      IF(K.EQ.KMAX) GO TO 1301                                                 00001520
      DO 1202 MN=1,MNMAXO                                                      00001530
      U1(MN)=U2(MN)                                                            00001540
      V1(MN)=V2(MN)                                                            00001550
      W1(MN)=W2(MN)                                                            00001560
      I3=3+(MN-1)*KMAX2                                                        00001570
      I2=I3-1                                                                  00001580
      U2(MN)=Z(1,I3)                                                           00001590
      V2(MN)=Z(2,I3)                                                           00001600
      W2(MN)=Z(3,I3)                                                           00001610
      PHIX(MN)=0.                                                              00001620
      PHIT(MN)=0.0                                                             00001630
      PHI(MN)=0.                                                               00001640
      MX(MN)=Z(4,I2)*ABZ3                                                      00001650
      MTH(MN)=0.                                                               00001660
      MXT(MN)=0.                                                               00001670
      QS(MN)=0.                                                                00001680
      TX(MN)=0.                                                                00001690
      TTH(MN)=0.                                                               00001700
 1202 TXT(MN)=0.                                                               00001710
      IF(M1.EQ.0) GO TO 1203                                                   00001720
      CALL INLPOL                                                              00001730
      M1L=M1-1                                                                 00001740
      M2L=M2-1                                                                 00001750
      PHIX(M1)=PHEE*ABZO                                                       00001760
      PHIX(M1L)=PHEN*ABZO                                                      00001770
      PHIT(M1)=-PHEE*ABZO                                                      00001780
```

```
      PHIT(M1L)=PHIX(M1L)                                                    00001790
      QS(M1)=0.                                                              00001800
      QS(M1L)=0.                                                             00001810
      IF(MO.EQ.0) GO TO 1204                                                 00001820
       TX(MO) =TO*ABZ                                                        00001830
      TTH(MO) =TO*ABZ                                                        00001840
      MTH(MO) =MX(MO)                                                        00001850
1204  IF(M2.EQ.0) GO TO 1205                                                 00001860
       TX(M2) = T2*ABZ                                                       00001870
      TTH(M2) =-T2*ABZ                                                       00001880
      TXT(M2)=-T2*ABZ                                                        00001890
      MTH(M2) =-MX(M2)                                                       00001900
      MXT(M2) =MTH(M2)                                                       00001910
      TX(M2L)=T2N*ABZ                                                        00001920
      TTH(M2L)=-TX(M2L)                                                      00001930
      TXT(M2L)=TX(M2L)                                                       00001940
      MTH(M2L)=-MX(M2L)                                                      00001950
      MXT(M2L)=MX(M2L)                                                       00001960
      GO TO 1205                                                             00001970
1203  IF(MO.EQ.0) GO TO 1206                                                 00001980
      I3=3+(MO-1)*KMAX2                                                      00001990
      I4=I3+1                                                                00002000
      CALL TLOAD(1)                                                          00002010
      TX(MO)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ       00002020
     1       -TT(MO)*ABZ*ALOAD                                               00002030
      TTH(MO)=TX(MO)                                                         00002040
      MTH(MO)=MX(MO)                                                         00002050
1206  IF(M2.EQ.0) GO TO 1205                                                 00002060
      I3=3+(M2-1)*KMAX2                                                      00002070
      I4=I3+1                                                                00002080
      TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL                               00002090
       TX(M2) = TX(M2)*ABZ                                                   00002100
      TTH(M2) =-TX(M2)                                                       00002110
      TXT(M2) =-TX(M2)                                                       00002120
      MTH(M2) =-MX(M2)                                                       00002130
      MXT(M2) =-MX(M2)                                                       00002140
      TX(M2L)=BS*D1*(2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))/DEL                  00002150
      TX(M2L)=TX(M2L)*ABZ                                                    00002160
      TTH(M2L)=-TX(M2L)                                                      00002170
      TXT(M2L)=TX(M2L)                                                       00002180
      MTH(M2L)=-MX(M2L)                                                      00002190
      MXT(M2L)=MX(M2L)                                                       00002200
1205  RETURN                                                                 00002210
1301  CONTINUE                                                               00002220
      DO 1302 MN=1,MNMAXO                                                    00002230
      U1(MN)=U2(MN)                                                          00002240
      V1(MN)=V2(MN)                                                          00002250
      W1(MN)=W2(MN)                                                          00002260
```

```
      PHIX(MN)=0.                                                                00002270
      PHIT(MN)=0.                                                                00002280
      PHI(MN)=0.                                                                 00002290
      IK=KMAX1+(MN-1)*KMAX2                                                      00002300
      MX(MN)=Z(4,IK)*ABZ3                                                        00002310
      MTH(MN)=0.                                                                 00002320
      MXT(MN)=0.                                                                 00002330
      QS(MN)=0.                                                                  00002340
      TX(MN)=0.                                                                  00002350
      TTH(MN)=0.                                                                 00002360
1302  TXT(MN)=0.                                                                 00002370
      IF(M1.EQ.0) GO TO 1303                                                     00002380
      CALL FNLPOL                                                                00002390
      PHIX(M1)=PHEE*ABZO                                                         00002400
      PHIX(M1L)=PHEN*ABZO                                                        00002410
      PHIT(M1)= PHEE*ABZO                                                        00002420
      PHIT(M1L)=-PHIX(M1L)                                                       00002430
      QS(M1)=0.                                                                  00002440
      QS(M1L)=0.                                                                 00002450
      IF(M0.EQ.0) GO TO 1304                                                     00002460
       TX(M0) =T0*ABZ                                                            00002470
      TTH(M0) =T0*ABZ                                                            00002480
      MTH(M0) =MX(M0)                                                            00002490
1304  IF(M2.EQ.0) GO TO 1305                                                     00002500
       TX(M2) = T2*ABZ                                                           00002510
      TTH(M2) =-T2*ABZ                                                           00002520
      TXT(M2) = T2*ABZ                                                           00002530
      MTH(M2) =-MX(M2)                                                           00002540
      MXT(M2) =MX(M2)                                                            00002550
      TX(M2L)=T2N*ABZ                                                            00002560
      TTH(M2L)=-TX(M2L)                                                          00002570
      TXT(M2L)=-TX(M2L)                                                          00002580
      MTH(M2L)=-MX(M2L)                                                          00002590
      MXT(M2L)=-MX(M2L)                                                          00002600
      GO TO 1305                                                                 00002610
1303  IF(M0.EQ.0) GO TO 1306                                                     00002620
      IKM=KMAX+(M0-1)*KMAX2                                                      00002630
      IM1=IKM-1                                                                  00002640
      CALL TLOAD(KMAX)                                                           00002650
      TX(M0)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+OMXI(KMAX)*Z(3,IKM+1)         00002660
     1          )*ABZ-TT(M0)*ABZ*ALOAD                                           00002670
      TTH(M0) =TX(M0)                                                            00002680
      MTH(M0) =MX(M0)                                                            00002690
1306  IF(M2.EQ.0) GO TO 1305                                                     00002700
      IKM=KMAX+(M2-1)*KMAX2                                                      00002710
      IM1=IKM-1                                                                  00002720
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL                                00002730
      TX(M2)=TX(M2)*ABZ                                                          00002740
```

```
      TTH(M2) =-TX(M2)                                                  00002750
      TXT(M2) = TX(M2)                                                  00002760
      MTH(M2) =-MX(M2)                                                  00002770
      MXT(M2) = MX(M2)                                                  00002780
      TX(M2L)=BS*D1*(-2.*Z(1,IKM-KMAX2)+.5*Z(1,IM1-KMAX2))/DEL          00002790
      TX(M2L)=TX(M2L)*ABZ                                               00002800
      TTH(M2L)=-TX(M2L)                                                 00002810
      TXT(M2L)=-TX(M2L)                                                 00002820
      MTH(M2L)=-MX(M2L)                                                 00002830
      MXT(M2L)=-MX(M2L)                                                 00002840
 1305 RETURN                                                            00002850
      END                                                               00002860
```



```
      SUBROUTINE OUTPUT(IMODE)                                          00000010
C*****************************************************************     00000020
C         THIS SUBROUTINE PREPARES THE PRINTOUT MATERIAL.  EVERY IPRINT *00000030
C         CONVERGED SOLUTION IS PRINTED. THE FOURIER COEFFICIENTS OF THE*00000040
C         INPLANE FORCES, MERIDIONAL TRANSVERSE FORCE, CIRCUMFERENTIAL  *00000050
C         BENDING MOMENT, TWISTING MOMENT AND ROTATIONS CAN BE COMPUTED *00000060
C         AND PRINTED WITH THE SOLUTION Z FOR THE FOURIER COEFFICIENTS  *00000070
C         OF THE THREE DISPLACEMENTS AND MERIDIONAL BENDING MOMENT. THIS*00000080
C         OUTPUT MATERIAL IS CONVERTED FROM DIMENSIONLESS FORM TO DIMEN-*00000090
C         SIONAL FORM HERE.  PROVISION IS MADE TO PRINT ONLY AT STATIONS*00000100
C         1, IFREQ+1, 2IFREQ+1, ETC.  THIS SUBROUTINE ALSO PERFORMS THE *00000110
C         SUMMATION PROCESS FOR COMPUTING THE TOTAL VALUES OF THE FORCES,00000120
C         MOMENTS, DISPLACEMENTS AND ROTATIONS AT THE NTHMAX POSITIONS  *00000130
C         AROUND THE CIRCUMFERENCE PRESCRIBED IN THE INPUT DATA.        *00000140
C*****************************************************************     00000150
      REAL NJ,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                 00000160
      INTEGER SORD                                                      00000170
      COMMON /IBL2/ N(25),MNINIT                                        00000180
      COMMON /IBL3/ M0,M1,M2,M3                                         00000190
      COMMON /IBL4/ KMAX,KL                                             00000200
      COMMON /IBL5/ IBCINL,IBCFNL                                       00000210
      COMMON /IBL7/ MNMAXO,MAXD(25),MAXS(25),MAXSY(25),IS(25,25),       00000220
     1              JS(25,25),ID(25,25),JD(25,25),IJS(25)               00000230
```

```
      COMMON /IBL8/ LSTEP,ITR                                                   00000240
      COMMON /IBL10/ IFREQ,NTHMAX                                               00000250
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS                                   00000260
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                          00000270
      COMMON /IBL13/ ITRMAX,LSMAX                                               00000280
      COMMON /BL4/  P(4,4,951),X(4,951),ZF1M(4,4,25),ZF2M(4,4,25),              00000290
     1              ZF3M(4,4,25),ZF4M(4,4,25)                                   00000300
      COMMON /BL5/  TT(25),MT(25),DT(25),DMT(25)                                00000310
      COMMON /BL6/  Z(4,1001),SOE,OSE,ALOAD                                     00000320
      COMMON /BL7/  D1,S1                                                       00000330
      COMMON /BL8/  R(100),GAM(100),OMT(100)                                    00000340
      COMMON /BL10/ PHIX(25),PHIT(25),PHI(25)                                   00000350
      COMMON /BL11/ OMXI(100),PHEE,T0,T2                                        00000360
      COMMON /BL12/ TDLI,TDEL                                                   00000370
      COMMON /BL14/ LAM2,LSD18,LSD1N                                            00000380
      COMMON /BL15/ NU,U1(25),U2(25),U3(25),V1(25),V2(25),V3(25),W1(25),        00000390
     1              W2(25),W3(25)                                               00000400
      COMMON /BL17/ DEL                                                         00000410
      COMMON /BL19/ TH(36)                                                      00000420
      COMMON /BL20/ DEOMX(100)                                                  00000430
      COMMON /BL27/ BX3(25),BT3(25),BXT3(25),BE3(25)                            00000440
      COMMON /BL31/ DELSQ,EXT1(25)                                              00000450
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                         00000460
      COMMON /BL100/ SORD,TEEO                                                  00000470
      COMMON /BL101/ ZO(4,1001),Z2(4,1001),Z3(4,1001),DELSD                     00000480
      COMMON /BL102/ DELOAD                                                     00000490
      COMMON /BL103/ MASS(100)                                                  00000500
      COMMON /BL110/ TX(25),TTH(25),TXT(25),MX(25),MTH(25),MXT(25),             00000510
     1               QS(25)                                                     00000520
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                                     00000530
      DIMENSION PTF(100),PF(100)                                                00000540
C*****************************************************************************00000550
      ABZO=SIGO/ELAST                                                           00000560
      IF(SORD.NE.0) GO TO 181                                                   00000570
      WRITE(6,101) LSTEP,ALOAD,ITR                                              00000580
      GO TO 182                                                                 00000590
  181 TI=LSTEP*DELOAD                                                           00000600
      DTI=TI*TEEO                                                               00000610
      WRITE(6,151) LSTEP,TI,DTI,ITR                                             00000620
  182 LAM=TKN/CHAR                                                              00000630
      ENL=1.                                                                    00000640
      ABZ=SIGO*TKN                                                              00000650
      ABZ3=ABZ*TKN*TKN/CHAR                                                     00000660
      ABZN=CHAR*SIGO/ELAST                                                      00000670
      IF(ITRMAX.EQ.1) ENL=0.                                                    00000680
      DD2=1.-NU**2                                                              00000690
      D2I=1./DD2                                                                00000700
      DPI=1./S1                                                                 00000710
```

```
      DNI=1./D1                                                                 00000720
      TDLSQI=.5/DELSQ                                                           00000730
      IF(NTHMAX.EQ.0) GO TO 991                                                 00000740
      DO 21 NTH=1,NTHMAX                                                        00000750
      DO 1 MN=1,MNMAXO                                                          00000760
      I1=1+(MN-1)*KMAX2                                                         00000770
      I2=I1+1                                                                   00000780
      U1(MN)=Z(1,I1)                                                            00000790
      U2(MN)=Z(1,I2)                                                            00000800
      V1(MN)=Z(2,I1)                                                            00000810
      V2(MN)=Z(2,I2)                                                            00000820
      W1(MN)=Z(3,I1)                                                            00000830
    1 W2(MN)=Z(3,I2)                                                            00000840
      THET=TH(NTH)                                                              00000850
      WRITE(6,116) THET                                                         00000860
      DO 121 K=1,KMAX                                                           00000870
      K1=K+1                                                                    00000880
      CALL BDB(K,BS,DB,DS,DD)                                                   00000890
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                                   00000900
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 999                                      00000910
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)                                00000920
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 999                                   00000930
      CALL PHIBET(K)                                                            00000940
      DEX=DEOMX(K)                                                              00000950
      RRA=1./R(K)                                                               00000960
      OX=OMXI(K)                                                                00000970
      OT=OMT(K)                                                                 00000980
      GA=GAM(K)                                                                 00000990
      DOXT=OX-OT                                                                00001000
      GDO=GA*DOXT                                                               00001010
      DD2D=DD2*DS                                                               00001020
      DO 3 MN=1,MNMAXO                                                          00001030
      EN=N(MN)                                                                  00001040
      ENR=EN*RRA                                                                00001050
      CALL TLOAD(K)                                                             00001060
      TTS=TT(MN)*ALOAD                                                          00001070
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))               00001080
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))               00001090
      EXT=0.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)                        00001100
     1     +ENL*SOE*BXT3(MN))                                                   00001110
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                               00001120
      KXT=0.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)        00001130
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DOXT*PHI(MN))                   00001140
      TX(MN)=BS*(EX+NU*ET)-TTS                                                  00001150
      TTH(MN)=BS*(ET+NU*EX)-TTS                                                 00001160
      TXT(MN)=BS*D1*EXT                                                         00001170
      MK1=K1+(MN-1)*KMAX2                                                       00001180
      MX(MN)=Z(4,MK1)                                                           00001190
```

```
      MTH(MN)=NU*MX(MN)+DD2D*KT-D1*MT(MN)*ALOAD                          00001200
      MXT(MN)=DS*D1*KXT                                                  00001210
      MK11=MK1+1                                                         00001220
      MKK1=MK1-1                                                         00001230
      QS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MK11)-Z(4,MKK1))*TDLI         00001240
     1        +ENR*MXT(MN)-GA*MTH(MN))                                    00001250
      MX(MN)=MX(MN)*ABZ3                                                 00001260
      MTH(MN)=MTH(MN)*ABZ3                                               00001270
      MXT(MN)=MXT(MN)*ABZ3                                               00001280
      TX(MN)=TX(MN)*ABZ                                                  00001290
      TTH(MN)=TTH(MN)*ABZ                                                00001300
      TXT(MN)=TXT(MN)*ABZ                                                00001310
      PHIX(MN)=PHIX(MN)*ABZO                                             00001320
      PHIT(MN)=PHIT(MN)*ABZO                                             00001330
      PHI(MN)=PHI(MN)*ABZO                                               00001340
      U1(MN)=U2(MN)                                                      00001350
      U2(MN)=U3(MN)                                                      00001360
      V1(MN)=V2(MN)                                                      00001370
      V2(MN)=V3(MN)                                                      00001380
      W1(MN)=W2(MN)                                                      00001390
    3 W2(MN)=W3(MN)                                                      00001400
      FK=K-1                                                             00001410
      FIFREQ=IFREQ                                                       00001420
      KTST=(K-1)/IFREQ                                                   00001430
      FKTST=KTST                                                         00001440
      FKTEST=FK/FIFREQ-FKTST                                             00001450
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999                                  00001460
      IF(FKTEST.NE.0.) GO TO 2                                           00001470
  999 X(1,K)=0.                                                          00001480
      X(2,K)=0.                                                          00001490
      X(3,K)=0.                                                          00001500
      X(4,K)=0.                                                          00001510
      PTF(K)=0.                                                          00001520
      PF(K)=0.                                                           00001530
      AMX=0.                                                             00001540
      AMTH=0.                                                            00001550
      AMXTH=0.                                                           00001560
      ANX=0.                                                             00001570
      ANTH=0.                                                            00001580
      ANXTH=0.                                                           00001590
      AQS=0.                                                             00001600
      IF (JUMP.EQ.2) GO TO 73                                            00001610
      DO 72 MN=1,MNMAXO                                                  00001620
      EN=N(MN)                                                           00001630
      FC=EN*THET                                                         00001640
      SN=SIN(FC)                                                         00001650
      CS=COS(FC)                                                         00001660
      X(1,K)=X(1,K)+U1(MN)*CS*ABZN                                       00001670
```

```
      X(2,K)=X(2,K)+V1(MN)*SN*ABZN                                          00001680
      X(3,K)=X(3,K)+W1(MN)*CS*ABZN                                          00001690
      X(4,K)=X(4,K)+PHIX(MN)*CS                                             00001700
      PTF(K)=PTF(K)+PHIT(MN)*SN                                             00001710
      AMX=AMX+MX(MN)*CS                                                     00001720
      AMTH=AMTH+MTH(MN)*CS                                                  00001730
      AMXTH=AMXTH+MXT(MN)*SN                                                00001740
      ANX=ANX+TX(MN)*CS                                                     00001750
      ANTH=ANTH+TTH(MN)*CS                                                  00001760
      AQS=AQS+QS(MN)*CS                                                     00001770
      ANXTH=ANXTH+TXT(MN)*SN                                                00001780
   72 PF(K)=PF(K)+PHI(MN)*SN                                                00001790
      GO TO 429                                                             00001800
   73 DO 74 MN=3,MNMAXO,JUMP                                                00001810
      EN=N(MN)                                                              00001820
      FC=EN*THET                                                            00001830
      SN=SIN(FC)                                                            00001840
      CS=COS(FC)                                                            00001850
      MNM=MN-1                                                              00001860
      X(1,K)=X(1,K)+(U1(MN)*CS+U1(MNM)*SN)*ABZN                             00001870
      X(2,K)=X(2,K)+(V1(MN)*SN+V1(MNM)*CS)*ABZN                             00001880
      X(3,K)=X(3,K)+(W1(MN)*CS+W1(MNM)*SN)*ABZN                             00001890
      X(4,K)=X(4,K)+PHIX(MN)*CS+PHIX(MNM)*SN                                00001900
      PTF(K)=PTF(K)+PHIT(MN)*SN+PHIT(MNM)*CS                                00001910
      AMX=AMX+MX(MN)*CS+MX(MNM)*SN                                          00001920
      AMTH=AMTH+MTH(MN)*CS+MTH(MNM)*SN                                      00001930
      AMXTH=AMXTH+MXT(MN)*SN+MXT(MNM)*CS                                    00001940
      ANX=ANX+TX(MN)*CS+TX(MNM)*SN                                          00001950
      ANTH=ANTH+TTH(MN)*CS+TTH(MNM)*SN                                      00001960
      ANXTH=ANXTH+TXT(MN)*SN+TXT(MNM)*CS                                    00001970
      AQS=AQS+QS(MN)*CS+QS(MNM)*SN                                          00001980
   74 PF(K)=PF(K)+PHI(MN)*SN+PHI(MNM)*CS                                    00001990
      X(1,K)=X(1,K)+U1(1)*ABZN                                              00002000
      X(2,K)=X(2,K)+V1(1)*ABZN                                              00002010
      X(3,K)=X(3,K)+W1(1)*ABZN                                              00002020
      X(4,K)=X(4,K)+PHIX(1)                                                 00002030
      PTF(K)=PTF(K)+PHIT(1)                                                 00002040
      PF(K)=PF(K)+PHI(1)                                                    00002050
      AMX=AMX+MX(1)                                                         00002060
      AMTH=AMTH+MTH(1)                                                      00002070
      AMXTH=AMXTH+MXT(1)                                                    00002080
      ANX=ANX+TX(1)                                                         00002090
      ANTH=ANTH+TTH(1)                                                      00002100
      ANXTH=ANXTH+TXT(1)                                                    00002110
      AQS=AQS+QS(1)                                                         00002120
  429 CONTINUE                                                              00002130
      IF(K.EQ.1) WRITE(6,117)                                               00002140
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))           00002150
```

```
    1 GO TO 220                                                                 00002160
      GO TO 221                                                                 00002170
  220 WRITE (6,1181) K,ANX,ANTH,ANXTH,AMX,AMTH,AMXTH                           00002180
      GO TO 2                                                                   00002190
  221 WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH                         00002200
    2 CONTINUE                                                                 00002210
  121 CONTINUE                                                                 00002220
      DO 660 K=1,KMAX                                                          00002230
      FK=K-1                                                                   00002240
      FIFREQ=IFREQ                                                             00002250
      KTST=(K-1)/IFREQ                                                         00002260
      FKTST=KTST                                                               00002270
      FKTEST=FK/FIFREQ-FKTST                                                   00002280
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 661                                        00002290
      IF(FKTEST.NE.0.) GO TO 658                                               00002300
  661 IF(K.EQ.1) WRITE(6,217)                                                  00002310
      WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),PTF(K),PF(K)                  00002320
  658 DO 659 I=1,4                                                             00002330
  659 X(I,K)=0.                                                                00002340
  660 CONTINUE                                                                 00002350
   21 CONTINUE                                                                 00002360
  991 IF(IMODE.LE.0) RETURN                                                    00002370
      DO 534 MN=1,MNMAXO                                                       00002380
      WRITE(6,749) N(MN)                                                       00002390
      DO 521 MM=1,MNMAXO                                                       00002400
      I1=1+(MM-1)*KMAX2                                                        00002410
      I2=I1+1                                                                  00002420
      U1(MM)=Z(1,I1)                                                           00002430
      U2(MM)=Z(1,I2)                                                           00002440
      V1(MM)=Z(2,I1)                                                           00002450
      V2(MM)=Z(2,I2)                                                           00002460
      W1(MM)=Z(3,I1)                                                           00002470
      W2(MM)=Z(3,I2)                                                           00002480
  521 CONTINUE                                                                 00002490
      DO 445 K=1, KMAX                                                         00002500
      K1=K+1                                                                   00002510
      CALL BDB(K,BS,DB,DS,DD)                                                  00002520
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                                  00002530
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)                               00002540
      TXZ=TX(MN)                                                               00002550
      TTHZ=TTH(MN)                                                             00002560
      TXTZ=TXT(MN)                                                             00002570
      AMXZ=MX(MN)                                                              00002580
      AMTHZ=MTH(MN)                                                            00002590
      AMXTZ=MXT(MN)                                                            00002600
      QSZ=QS(MN)                                                               00002610
      X(1,K)=PHIX(MN)                                                          00002620
      X(2,K)=PHIT(MN)                                                          00002630
```

```
      X(3,K)=PHI(MN)                                                                00002640
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 583                                          00002650
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 583                                       00002660
      CALL PHIBET(K)                                                                00002670
      DEX=DEOMX(K)                                                                  00002680
      RRA=1./R(K)                                                                   00002690
      OX=OMXI(K)                                                                    00002700
      OT=OMT(K)                                                                     00002710
      GA=GAM(K)                                                                     00002720
      DOXT=OX-OT                                                                    00002730
      GDO=GA*DOXT                                                                   00002740
      DD2D=DD2*DS                                                                   00002750
      EN=N(MN)                                                                      00002760
      ENR=EN*RRA                                                                    00002770
      CALL TLOAD(K)                                                                 00002780
      TTS=TT(MN)*ALOAD                                                              00002790
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))                   00002800
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))                   00002810
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)                             00002820
     1     +ENL*SOE*BXT3(MN))                                                       00002830
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                                   00002840
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)             00002850
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DOXT*PHI(MN))                       00002860
      TXZ=(BS*(EX+NU*ET)-TTS)*ABZ                                                   00002870
      TTHZ=(BS*(ET+NU*EX)-TTS)*ABZ                                                  00002880
      TXTZ=BS*D1*EXT*ABZ                                                            00002890
      MK1=K1+(MN-1)*KMAX2                                                           00002900
      AMXZ=Z(4,MK1)                                                                 00002910
      AMTHZ=NU*AMXZ+DD2D*KT-D1*MT(MN)*ALOAD                                         00002920
      AMXTZ=DS*D1*KXT                                                               00002930
      MK11=MK1+1                                                                    00002940
      MKK1=MK1-1                                                                    00002950
      QSZ=SIGO*TKN*LAM2*(GA*AMXZ+(Z(4,MK11)-Z(4,MKK1))*TDLI                         00002960
     1     +ENR*AMXTZ-GA*AMTHZ)                                                     00002970
      AMXZ=AMXZ*ABZ3                                                                00002980
      AMTHZ=AMTHZ*ABZ3                                                              00002990
      AMXTZ=AMXTZ*ABZ3                                                              00003000
      X(1,K)=PHIX(MN)*ABZO                                                          00003010
      X(2,K)=PHIT(MN)*ABZO                                                          00003020
      X(3,K)=PHI(MN)*ABZO                                                           00003030
      DO 533 MM=1,MNMAXO                                                            00003040
      U1(MM)=U2(MM)                                                                 00003050
      U2(MM)=U3(MM)                                                                 00003060
      V1(MM)=V2(MM)                                                                 00003070
      V2(MM)=V3(MM)                                                                 00003080
      W1(MM)=W2(MM)                                                                 00003090
  533 W2(MM)=W3(MM)                                                                 00003100
      FK=K-1                                                                        00003110
```

```
      FIFREQ=IFREQ                                                          00003120
      KTST=(K-1)/IFREQ                                                      00003130
      FKTST=KTST                                                            00003140
      FKTEST=FK/FIFREQ-FKTST                                                00003150
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583                                     00003160
      IF(FKTEST.NE.0.) GO TO 445                                            00003170
  583 CONTINUE                                                              00003180
      IF(K.EQ.1) WRITE(6,117)                                               00003190
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))           00003200
     1 GO TO 4444                                                           00003210
      GO TO 4445                                                            00003220
 4444 WRITE (6,1181) K,TXZ,TTHZ,TXTZ,AMXZ,AMTHZ,AMXTZ                      00003230
      GO TO 445                                                             00003240
 4445 WRITE(6,118) K,TXZ,TTHZ,TXTZ,QSZ,AMXZ,AMTHZ,AMXTZ                    00003250
  445 CONTINUE                                                              00003260
      WRITE(6,217)                                                          00003270
      DO 446 K=1,KMAX                                                       00003280
      FK=K-1                                                                00003290
      FIFREQ=IFREQ                                                          00003300
      KTST=(K-1)/IFREQ                                                      00003310
      FKTST=KTST                                                            00003320
      FKTEST=FK/FIFREQ-FKTST                                                00003330
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 593                                     00003340
      IF(FKTEST.NE.0.) GO TO 446                                            00003350
  593 KZ=K+1+(MN-1)*KMAX2                                                   00003360
      UP=Z(1,KZ)*ABZN                                                       00003370
      VP=Z(2,KZ)*ABZN                                                       00003380
      WP=Z(3,KZ)*ABZN                                                       00003390
      WRITE(6,218) K,UP,VP,WP,X(1,K),X(2,K),X(3,K)                          00003400
  446 CONTINUE                                                              00003410
  534 CONTINUE                                                              00003420
C*****************************************************************        00003430
  101 FORMAT(1H1,34H          THE LOAD STEP NUMBER IS I2,29H          TH00003440
     1E LOAD FACTOR IS E11.4,36H          THE SOLUTION CONVERGED IN I2,10000003450
     21H ITERATIONS////)                                                    00003460
  102 FORMAT(1H0,35H          MODAL OUTPUT FOR STATION I3//)                00003470
  103 FORMAT(//7H   N   16H       U         16H       V         16H        00003480
     1W         16H       PHI S    16H       PHI THETA10H       PHI/)     00003490
  104 FORMAT(2X,I2,3X,6E16.4)                                               00003500
  105 FORMAT(7H   N   16H       N S      16H       N THETA  16H       N 00003510
     1STHETA 16H       Q S      16H       M S      16H       M THETA  1600003520
     2H       M STHETA /)                                                  00003530
  106 FORMAT(1H0,8HSTATION I3,14H- THETA OUTPUT/ )                          00003540
  107 FORMAT(                  14H     THETA    14H     U          14H     00003550
     1 V         14H     W         14H     PHI S    14H     PHI THETA14H  00003560
     2   PHI      )                                                        00003570
  108 FORMAT(8E14.4)                                                        00003580
  109 FORMAT(1H0,                14H     THETA    14H    M S        14H 00003590
```

```
     1   M THETA   14H    M STHETA  14H    N S       14H    N THETA   140000360
     2H    N STHETA  14H     Q S       )                                00003610
  114 FORMAT(2X,I2,3X,7E16.4)                                           00003620
  115 FORMAT(7H    N   16H        N S       16H       N THETA  16H        N 00003630
     1STHETA 16H        M S      16H        M THETA  15H        M STHETA//)00003640
  116 FORMAT(1HO,84H          THE SUMMED FORCES, MOMENTS, DISPLACEMENTS 00003650
     1AND ROTATIONS FOLLOW FOR THETA =E15.6///)                         00003660
  117 FORMAT(/,8H STATION15H       N S       16H       N THETA  16H      00003670
     1N STHETA 16H        Q S       16H       M S       16H       M THETA  00003680
     216H       M STHETA //)                                           00003690
  118 FORMAT(1X,I3,3X,7E16.4)                                           00003700
 1181 FORMAT(1X,I3,3X,3E16.4,'  NOT COMPUTED  ',3E16.4)                 00003710
  151 FORMAT(1H1,26H  THE TIME STEP NUMBER IS I4,18H      THE TIME IS F500003720
     1.2,4H OR E9.3,40H SECONDS      THE SOLUTION CONVERGED IN I2,11H IT00003730
     3ERATIONS////)                                                     00003740
  217 FORMAT(//8H STATION,15H       U         16H       V         16H    00003750
     1 W         16H       PHI S    16H       PHI THETA16H       PHI    00003760
     2 /)                                                               00003770
  218 FORMAT(1X,I3,3X,6E16.4)                                           00003780
  749 FORMAT(1H1,40X,27H MODAL OUTPUT FOR MODE N = I3,8H FOLLOWS)       00003790
      RETURN                                                            00003800
      END                                                               00003810
                                                                        00003820
```

```
      SUBROUTINE IMPERF
      COMMON /IBL11/ ICORFL,IPASS,JUMP,MPERFS
      COMMON /BLIMP/ PHIXB(951),PHITB(951)
      DIMENSION FNT(25),DFNT(25)
C*****************************************************************
C     THIS SUBROUTINE EITHER CAN COMPUTE THE INITIAL IMPERFECTIONS OF  *
C     THE SHELL AND INITIALIZE ARRAYS PHIXB AND PHITB, OR JUST CAUSE   *
C     A LOADING OF THE ARRAYS FROM PRE-PREPARED DATA CARDS.  IF 'MPERFS*
C     IS 0, THE ARRAYS PHIXB AND PHITB ARE MADE ZERO.                  *
C     NOTE: WHEN LOADING ARRAYS PHIXB AND PHITB WITH THE IMPERFECTION  *
C     ROTATION FOURIER COEFFICIENTS, THE FOLLOWING ORDERING MUST BE    *
C     FOLLOWED:     FIRST KMAX LOCATIONS CONTAIN MODE '0',             *
C                   SECOND KMAX LOCATIONS CONTAIN MODE '-1',           *
```

```
C                    THIRD KMAX LOCATIONS CONTAIN MODE '+I',                *
C                    FOURTH KMAX LOCATIONS CONTAIN MODE '-J',               *
C                    FIFTH KMAX LOCATIONS CONTAIN MODE '+J',                *
C                    SIXTH KMAX LOCATIONS CONTAIN MODE '-K', ETC.           *
C*****************************************************************************
      IF (MPERFS.EQ.1) GO TO 2
      DO 1 I=1,951
      PHITB(I)=0.0
    1 PHIXB(I)=0.0
      GO TO 3
    2 CONTINUE
C*****************************************************************************
C     FROM HERE TO THE NEXT ROW OF ASTERISKS IS WHERE THE COEFFICIENTS
C     ARE EITHER COMPUTED AND READ, OR JUST READ.
      READ (5,1002)(PHIXB(I),I=1,875)
      READ (5,1002)(PHITB(I),I=1,875)
 1002 FORMAT (174(5E14.6/),5E14.6)
C     FOLLOWING CARDS REDUCE IMPERFECTIONS TO 'MINUTE' SIZE.
      DO 7 I=1,875
      PHIXB(I)=PHIXB(I)*0.0001
    7 PHITB(I)=PHITB(I)*0.0001
C*****************************************************************************
    3 RETURN
      END
```

APPENDIX G

LISTING OF IMPERFECTION DATA HANDLING PROGRAM

```
C      PROGRAM TO PREPARE RAW IMPERFECTION DATA FOR USE BY SATANS-II      00000010
       DIMENSION A14(1422),AM14(35,49),PHIXRO(35,49),PHITRO(35,49),       00000020
      1          TEMP(48),A(25),B(25),PXROCC(35,25),PXROSC(35,25),        00000030
      2          PTROCC(35,25),PTROSC(35,25),PXRCCS(25),PXRSCS(25),       00000040
      3          PTRCCS(25),PTRSCS(25),CHECK(361),C(25),S(25),NN(13),     00000050
      4          PHIXB(951),PHITB(951),E(25)                              00000060
C                                                                         00000070
CC     READ RAW DATA INTO COLUMN MATRIX, A14:                             00000080
C                                                                         00000090
       DO 1 I=1,237                                                       00000100
       K=6*(I-1)                                                          00000110
     1 READ (5,1000) A14(K+1),A14(K+2),A14(K+3),A14(K+4),A14(K+5),        00000120
      1              A14(K+6)                                            00000130
C                                                                         00000140
CCCC    TRANSFER DATA INTO 35X49 MATRIX: (NOTE - ROWS 1,2,3,33,34,35 ARE  00000150
       UNKNOWN AT THIS POINT)                                             00000160
CCCC   IN THE PROCESS, DIMENSIONALIZE THE DATA (MULT. BY SHELL THICKNESS)00000170
C                                                                         00000180
       K=0                                                                00000190
       DO 2 I=4,32                                                        00000200
       DO 2 J=1,49                                                        00000210
       K=K+1                                                              00000220
     2 AM14(I,J)=A14(K)*0.00437                                           00000230
C                                                                         00000240
CCC    PERFORM MODIFIED LINEAR EXTRAPOLATION FOR END POINTS               00000250
C                                                                         00000260
       DO 3 J=1,49                                                        00000270
       D1=(AM14(4,J)+AM14(5,J))/2.0                                       00000280
       D2=(AM14(6,J)+AM14(7,J))/2.0                                       00000290
       D3=(AM14(32,J)+AM14(31,J))/2.0                                     00000300
       D4=(AM14(30,J)+AM14(29,J))/2.0                                     00000310
       THETAT=(D1-D2)/0.48                                                00000320
       THETAB=(D3-D4)/0.48                                                00000330
       AM14(3,J)=D1+THETAT*0.36                                           00000340
       AM14(2,J)=D1+THETAT*0.6                                            00000350
       AM14(1,J)=D1+THETAT*0.84                                           00000360
       AM14(33,J)=D3+THETAB*0.36                                          00000370
       AM14(34,J)=D3+THETAB*0.6                                           00000380
     3 AM14(35,J)=D3+THETAB*0.84                                          00000390
C                                                                         00000400
CCC    FULL 35X49 MATRIX OF DIMENSIONAL DEVIATIONS EXISTS.                00000410
C                                                                         00000420
```

```
      WRITE (6,1001)                                                    00000430
      WRITE (6,1002)                                                    00000440
      WRITE (6,1003) ((AM14(I,J),J=1,49),I=1,35)                        00000450
C                                                                       00000460
CCC   COMPUTE MERIDIONAL ROTATIONS USING FINITE DIFFERENCING            00000470
C                                                                       00000480
      DO 4 J=1,49                                                       00000490
      DO 5 I=2,34                                                       00000500
      IM1=I-1                                                           00000510
      IP1=I+1                                                           00000520
    5 PHIXRO(I,J)=(AM14(IM1,J)-AM14(IP1,J))/0.48                        00000530
      PHIXRO(1,J)=PHIXRO(2,J)                                           00000540
    4 PHIXRO(35,J)=PHIXRO(34,J)                                         00000550
C                                                                       00000560
CCC   COMPUTE CIRCUMFERENTIAL ROTATIONS IN SIMILAR FASHION              00000570
CCC   NOTE: SPACING EQUALS CIRCUMFERENCE/INCREMENTS = PI/6)             00000580
C                                                                       00000590
      DO 6 I=1,35                                                       00000600
      DO 7 J=2,48                                                       00000610
      JM1=J-1                                                           00000620
      JP1=J+1                                                           00000630
    7 PHITRO(I,J)=(AM14(I,JM1)-AM14(I,JP1))/1.047198                    00000640
      PHITRO(I,1)=(AM14(I,48)-AM14(I,2))/1.047198                       00000650
    6 PHITRO(I,49)=PHITRO(I,1)                                          00000660
C                                                                       00000670
CCC   FULL 35X49 MATRICES OF MERIDIONAL AND CIRCUMFERENTIAL ROTATIONS   00000680
CCC   EXIST.                                                            00000690
C                                                                       00000700
      WRITE (6,1001)                                                    00000710
      WRITE (6,1004)                                                    00000720
      WRITE (6,1003) ((PHIXRO(I,J),J=1,49),I=1,35)                      00000730
      WRITE (6,1001)                                                    00000740
      WRITE (6,1005)                                                    00000750
      WRITE (6,1003) ((PHITRO(I,J),J=1,49),I=1,35)                      00000760
C                                                                       00000770
CCC   NONDIMENSIONALIZE THE PHIXRO AND PHITRO MATRICES FOR USE BY       00000780
CCC   SATANS-II - MULTIPLY BY EO/SIGO                                   00000790
C                                                                       00000800
      DO 77 I=1,35                                                      00000810
      DO 77 J=1,49                                                      00000820
      PHIXRO(I,J)=PHIXRO(I,J)*0.158E5                                   00000830
   77 PHITRO(I,J)=PHITRO(I,J)*0.158E5                                   00000840
      WRITE (6,1001)                                                    00000850
      WRITE (6,2004)                                                    00000860
      WRITE (6,1003) ((PHIXRO(I,J),J=1,49),I=1,35)                      00000870
      WRITE (6,1001)                                                    00000880
      WRITE (6,2005)                                                    00000890
      WRITE (6,1003) ((PHITRO(I,J),J=1,49),I=1,35)                      00000900
```

```
C                                                                       00000910
CCC     NOW EXPAND EACH CIRCUMFERENCE OF PHIXRO'S AND PHITRO'S INTO 49   00000920
CCCC    ELEMENT FOURIER SERIES.  PUT SINE AND COSINE COEFFS. INTO ARRAYS 00000930
CCC     PXROCC,PXROSC,PTROCC,PTROSC.                                    00000940
C                                                                       00000950
        DO 8 I=1,35                                                     00000960
        DO 9 J=1,48                                                     00000970
      9 TEMP(J)=PHIXRO(I,J)                                             00000980
        CALL FORIER (48,TEMP,25,AHO,SYS,A,B,E)                          00000990
        DO 901 KK=1,24                                                  00001000
        K=25-KK                                                         00001010
        KP1=K+1                                                         00001020
        A(KP1)=A(K)                                                     00001030
    901 B(KP1)=B(K)                                                     00001040
        A(1)=AHO                                                        00001050
        B(1)=0.0                                                        00001060
        DO 10 J=1,25                                                    00001070
        PXROCC(I,J)=A(J)                                                00001080
     10 PXROSC(I,J)=B(J)                                                00001090
        DO 11 J=1,48                                                    00001100
     11 TEMP(J)=PHITRO(I,J)                                             00001110
        CALL FORIER (48,TEMP,25,AHO,SYS,A,B,E)                          00001120
        DO 902 KK=1,24                                                  00001130
        K=25-KK                                                         00001140
        KP1=K+1                                                         00001150
        A(KP1)=A(K)                                                     00001160
    902 B(KP1)=B(K)                                                     00001170
        A(1)=AHO                                                        00001180
        B(1)=0.0                                                        00001190
        DO 12 J=1,25                                                    00001200
        PTROCC(I,J)=A(J)                                                00001210
     12 PTROSC(I,J)=B(J)                                                00001220
      8 CONTINUE                                                        00001230
C                                                                       00001240
CCC     A 49-ELEMENT FOURIER EXPANSION FOR MERIDIONAL AND CIRCUMFERENTIAL 00001250
CCC     ROTATION IMPERFECTIONS EXISTS FOR EACH CIRCUMFERENCE.           00001260
C                                                                       00001270
        WRITE (6,1001)                                                  00001280
        WRITE (6,1006)                                                  00001290
        DO 13 I=1,35                                                    00001300
     13 WRITE (6,1007) (PXROCC(I,J),J=1,25),(PXROSC(I,J),J=1,25)        00001310
        WRITE (6,1001)                                                  00001320
        WRITE (6,1008)                                                  00001330
        DO 14 I=1,35                                                    00001340
     14 WRITE (6,1007) (PTROCC(I,J),J=1,25),(PTROSC(I,J),J=1,25)        00001350
C                                                                       00001360
CCC     DETERMINE RELATIVE IMPORTANCE OF EACH MODAL COMPONENT OF THE MERI-00001370
C       DIONAL AND CIRCUMFERENTIAL ROTATIONS                            00001380
```

```
C                                                                       00001390
      DO 15 J=1,25                                                      00001400
      PXCSUM=0.0                                                        00001410
      PXSSUM=0.0                                                        00001420
      PTCSUM=0.0                                                        00001430
      PTSSUM=0.0                                                        00001440
      DO 16 I=1,35                                                      00001450
      PXCSUM=PXCSUM+ABS(PXROCC(I,J))                                    00001460
      PXSSUM=PXSSUM+ABS(PXROSC(I,J))                                    00001470
      PTCSUM=PTCSUM+ABS(PTROCC(I,J))                                    00001480
   16 PTSSUM=PTSSUM+ABS(PTROSC(I,J))                                    00001490
      PXRCCS(J)=PXCSUM/35.0                                             00001500
      PXRSCS(J)=PXSSUM/35.0                                             00001510
      PTRCCS(J)=PTCSUM/35.0                                             00001520
   15 PTRSCS(J)=PTSSUM/35.0                                             00001530
      WRITE (6,1001)                                                    00001540
      WRITE (6,1009)                                                    00001550
      DO 17 I=1,25                                                      00001560
      J=I-1                                                             00001570
   17 WRITE (6,1010) J,PXRCCS(I),PXRSCS(I),PTRCCS(I),PTRSCS(I)          00001580
C                                                                       00001590
CCCC  PICK CIRCUMFERENCE 10 AND COMPARE ACTUAL VS FOURIER REPRESENTATION00001600
CCCC  OF CIRCUMFERENTIAL ROTATION IMPERFECTION.                         00001610
C                                                                       00001620
      DO 18 I=1,25                                                      00001630
      A(I)=PTROCC(10,I)                                                 00001640
   18 B(I)=PTROSC(10,I)                                                 00001650
      STEP=.1745329E-1                                                  00001660
      DO 19 I=1,361                                                     00001670
      THETA=FLOAT(I-1)*STEP                                             00001680
      DO 20 J=2,25                                                      00001690
      RJ=FLOAT(J-1)                                                     00001700
      C(J)=A(J)*COS(RJ*THETA)                                           00001710
   20 S(J)=B(J)*SIN(RJ*THETA)                                           00001720
      CHECK(I)=A(1)                                                     00001730
      DO 21 J=2,25                                                      00001740
   21 CHECK(I)=CHECK(I)+C(J)+S(J)                                       00001750
   19 CONTINUE                                                          00001760
      WRITE (6,1001)                                                    00001770
      WRITE (6,1011)                                                    00001780
      K=1                                                               00001790
      DO 22 I=1,361,15                                                  00001800
      II=I-1                                                            00001810
      WRITE (6,1012) II,PHITRO(10,K),CHECK(I)                           00001820
   22 K=K+2                                                             00001830
      WRITE (6,1013)                                                    00001840
      DO 23 I=1,90                                                      00001850
      I1=I-1                                                            00001860
```

```
      I2=I1+90                                                          00001870
      I3=I1+180                                                         00001880
      I4=I1+270                                                         00001890
   23 WRITE (6,1014) I1,CHECK(I),I2,CHECK(I+90),I3,CHECK(I+180),I4,     00001900
     1                CHECK(I+270)                                     00001910
C                                                                       00001920
CC    IF MODES FOR USE BY SATANS-II ARE KNOWN REMOVE 'GO-TO' AND PROCEED00001930
CC    GO TO 9999                                                        00001940
CC                                                                      00001950
CC     READ MODES TO BE USED IN SATANS-II ANALYSIS                      00001960
CC    NOTE: MODES MUST INCLUDE MODE '0',                                00001970
C                                                                       00001980
  231 READ (5,1015) (NN(I),I=1,13)                                      00001990
      WRITE (6,1001)                                                    00002000
      WRITE (6,1020) (NN(I),I=1,13)                                     00002010
C                                                                       00002020
CC    SINCE MODE '0' LIVES IN SLOT '1', ETC., BOUNCE EACH VALUE 1.      00002030
CC                                                                      00002040
      DO 24 I=1,13                                                      00002050
   24 NN(I)=NN(I)+1                                                     00002060
C                                                                       00002070
CC    SEE HOW CIRC. 10 LOOKS WITH THESE SELECTED MODES.                 00002080
      DO 25 I=1,361                                                     00002090
      THETA=FLOAT(I-1)*STEP                                             00002100
      DO 26 J=2,13                                                      00002110
      RJ=FLOAT(NN(J)-1)                                                 00002120
      C(NN(J))=A(NN(J))*COS(RJ*THETA)                                   00002130
   26 S(NN(J))=B(NN(J))*SIN(RJ*THETA)                                   00002140
      CHECK(I)=A(1)                                                     00002150
      DO 27 J=2,13                                                      00002160
   27 CHECK(I)=CHECK(I)+C(NN(J))+S(NN(J))                               00002170
   25 CONTINUE                                                          00002180
      WRITE (6,1013)                                                    00002190
      DO 28 I=1,90                                                      00002200
      I1=I-1                                                            00002210
      I2=I1+90                                                          00002220
      I3=I1+180                                                         00002230
      I4=I1+270                                                         00002240
   28 WRITE (6,1014) I1,CHECK(I),I2,CHECK(I+90),I3,CHECK(I+180),I4,     00002250
     1                CHECK(I+270)                                     00002260
C                                                                       00002270
CC    PUT APPROPRIATE ROTATION IMPERFECTION COEFFS INTO ARRAYS 'PHIXB'  00002280
CC    AND 'PHITB' FOR USE BY SATANS-II.                                 00002290
CC    NOTE: ORDER IN WHICH SATANS-II USES PHIXB & PHITB IS AS FOLLOWS:  00002300
CC    FIRST KMAX LOCNS ARE MODE '0'                                     00002310
CC    SECOND KMAX LOCNS ARE MODE '-I'                                   00002320
CC    THIRD KMAX LOCNS ARE MODE 'I'                                     00002330
C     FOURTH KMAX LOCNS ARE MODE '-J'                                   00002340
```

```
C      FIFTH KMAX LOCNS ARE MODE 'J'.                                    00002350
CC     SIXTH KMAX LOCNS ARE MODE '-K', ETC.....                          00002360
C                                                                        00002370
       DO 101 I=1,35                                                     00002380
       PHIXB(I)=PXROCC(I,1)                                              00002390
  101  PHITB(I)=PTROCC(I,1)                                              00002400
       DO 102 I=1,35                                                     00002410
       II=I+70                                                           00002420
       JJ=II-35                                                          00002430
       PHIXB(II)=PXROCC(I,NN(2))                                         00002440
       PHITB(II)=PTROCC(I,NN(2))                                         00002450
       PHIXB(JJ)=PXROSC(I,NN(2))                                         00002460
  102  PHITB(JJ)=PTROSC(I,NN(2))                                         00002470
       DO 103 I=1,35                                                     00002480
       II=I+140                                                          00002490
       JJ=II-35                                                          00002500
       PHIXB(II)=PXROCC(I,NN(3))                                         00002510
       PHITB(II)=PTROCC(I,NN(3))                                         00002520
       PHIXB(JJ)=PXROSC(I,NN(3))                                         00002530
  103  PHITB(JJ)=PTROSC(I,NN(3))                                         00002540
       DO 104 I=1,35                                                     00002550
       II=I+210                                                          00002560
       JJ=II-35                                                          00002570
       PHIXB(II)=PXROCC(I,NN(4))                                         00002580
       PHITB(II)=PTROCC(I,NN(4))                                         00002590
       PHIXB(JJ)=PXROSC(I,NN(4))                                         00002600
  104  PHITB(JJ)=PTROSC(I,NN(4))                                         00002610
       DO 105 I=1,35                                                     00002620
       II=I+280                                                          00002630
       JJ=II-35                                                          00002640
       PHIXB(II)=PXROCC(I,NN(5))                                         00002650
       PHITB(II)=PTROCC(I,NN(5))                                         00002660
       PHIXB(JJ)=PXROSC(I,NN(5))                                         00002670
  105  PHITB(JJ)=PTROSC(I,NN(5))                                         00002680
       DO 106 I=1,35                                                     00002690
       II=I+350                                                          00002700
       JJ=II-35                                                          00002710
       PHIXB(II)=PXROCC(I,NN(6))                                         00002720
       PHITB(II)=PTROCC(I,NN(6))                                         00002730
       PHIXB(JJ)=PXROSC(I,NN(6))                                         00002740
  106  PHITB(JJ)=PTROSC(I,NN(6))                                         00002750
       DO 107 I=1,35                                                     00002760
       II=I+420                                                          00002770
       JJ=II-35                                                          00002780
       PHIXB(II)=PXROCC(I,NN(7))                                         00002790
       PHITB(II)=PTROCC(I,NN(7))                                         00002800
       PHIXB(JJ)=PXROSC(I,NN(7))                                         00002810
  107  PHITB(JJ)=PTROSC(I,NN(7))                                         00002820
```

```
      DO 108 I=1,35                                                    00002830
      II=I+490                                                         00002840
      JJ=II-35                                                         00002850
      PHIXB(II)=PXROCC(I,NN(8))                                        00002860
      PHITB(II)=PTROCC(I,NN(8))                                        00002870
      PHIXB(JJ)=PXROSC(I,NN(8))                                        00002880
  108 PHITB(JJ)=PTROSC(I,NN(8))                                        00002890
      DO 109 I=1,35                                                    00002900
      II=I+560                                                         00002910
      JJ=II-35                                                         00002920
      PHIXB(II)=PXROCC(I,NN(9))                                        00002930
      PHITB(II)=PTROCC(I,NN(9))                                        00002940
      PHIXB(JJ)=PXROSC(I,NN(9))                                        00002950
  109 PHITB(JJ)=PTROSC(I,NN(9))                                        00002960
      DO 110 I=1,35                                                    00002970
      II=I+630                                                         00002980
      JJ=II-35                                                         00002990
      PHIXB(II)=PXROCC(I,NN(10))                                       00003000
      PHITB(II)=PTROCC(I,NN(10))                                       00003010
      PHIXB(JJ)=PXROSC(I,NN(10))                                       00003020
  110 PHITB(JJ)=PTROSC(I,NN(10))                                       00003030
      DO 111 I=1,35                                                    00003040
      II=I+700                                                         00003050
      JJ=II-35                                                         00003060
      PHIXB(II)=PXROCC(I,NN(11))                                       00003070
      PHITB(II)=PTROCC(I,NN(11))                                       00003080
      PHIXB(JJ)=PXROSC(I,NN(11))                                       00003090
  111 PHITB(JJ)=PTROSC(I,NN(11))                                       00003100
      DO 112 I=1,35                                                    00003110
      II=I+770                                                         00003120
      JJ=II-35                                                         00003130
      PHIXB(II)=PXROCC(I,NN(12))                                       00003140
      PHITB(II)=PTROCC(I,NN(12))                                       00003150
      PHIXB(JJ)=PXROSC(I,NN(12))                                       00003160
  112 PHITB(JJ)=PTROSC(I,NN(12))                                       00003170
      DO 113 I=1,35                                                    00003180
      II=I+840                                                         00003190
      JJ=II-35                                                         00003200
      PHIXB(II)=PXROCC(I,NN(13))                                       00003210
      PHITB(II)=PTROCC(I,NN(13))                                       00003220
      PHIXB(JJ)=PXROSC(I,NN(13))                                       00003230
  113 PHITB(JJ)=PTROSC(I,NN(13))                                       00003240
      WRITE (6,1001)                                                   00003250
      WRITE (6,1016)                                                   00003260
      WRITE (6,1017) (PHIXB(I),I=1,875)                                00003270
      WRITE (6,1001)                                                   00003280
      WRITE (6,1018)                                                   00003290
      WRITE (6,1017) (PHITB(I),I=1,875)                                00003300
```

```
C                                                                       00003310
C     IF PHIXB AND PHITB LOOK GOOD, REMOVE 'GO-TO' AND PUNCH OUT DATA   00003320
C     GO TO 9999                                                        00003330
C                                                                       00003340
  250 WRITE (7,1019) (PHIXB(I),I=1,875)                                 00003350
      WRITE (7,1019) (PHITB(I),I=1,875)                                 00003360
C                                                                       00003370
 1000 FORMAT (6E12.5)                                                   00003380
 1001 FORMAT ('1')                                                      00003390
 1002 FORMAT (' ','DIMENSIONAL DEVIATIONS, READ LEFT TO RIGHT 5 LINES FO00003400
     1R EACH CIRCUMFERENTIAL TRACE, 35 SUCH TRACES'//)                  00003410
 1003 FORMAT      (35(' ',10F13.8/' ',10F13.8/' ',10F13.8/' ',10F13.8/' '00003420
     1,9F13.8//))                                                       00003430
 1004 FORMAT (' ','MERIDIONAL ROTATIONS (IN RADIANS), READ LEFT TO RIGHT00003440
     1 5-LINES FOR EACH CIRCUMFERENCE'//)                               00003450
 2004 FORMAT (' ','MERIDIONAL ROTATIONS (NONDIMENSNL) READ LEFT TO RIGHT00003460
     1 5-LINES FOR EACH CIRCUMFERENCE'//)                               00003470
 1005 FORMAT (' ','CIRCUMFERENTIAL ROTATIONS (IN RADIANS), READ LEFT TO 00003480
     1RIGHT 5 LINES FOR EACH CIRCUMFERENCE'//)                          00003490
 2005 FORMAT (' ','CIRCUMFERENTIAL ROTATIONS (NONDIMENSNL) READ LEFT TO 00003500
     1RIGHT 5 LINES FOR EACH CIRCUMFERENCE'//)                          00003510
 1006 FORMAT (' ','FOURIER COEFFICIENTS FOR MERIDIONAL ROTATIONS, READ (00003520
     1FOR EA. CIRCUM.) 3 LINES OF 25 COS COEFFS, 3 LINES OF 25 SIN COEFF00003530
     2S.'//)                                                            00003540
 1007 FORMAT      (2(' ',10E13.6/' ',10E13.6/' ',5E13.6/)//)            00003550
 1008 FORMAT (' ','FOURIER COEFFICIENTS FOR CIRCUMFERENTIAL ROTATIONS, R00003560
     1EAD AS BEFORE'//)                                                 00003570
 1009 FORMAT (' ','SHELL-14, IMPERFECTION ROTATION FOURIER COEFFICIENT R00003580
     1ELATIVE AVERAGE SIZE'//' ',T10,'MODE',T20,'PXRO COS COEFF',T40,'PX00003590
     2RO SIN COEFF',T60,'PTRO COS COEFF',T80,'PTRO SIN COEFF'//)         00003600
 1010 FORMAT (' ',T11,I3,T20,F12.6,T40,F12.6,T60,F12.6,T80,F12.6)       00003610
 1011 FORMAT (' ',T10,'DEGREE',T20,'ACTUAL PHITRO',T40,'FOURIER PHITRO')00003620
 1012 FORMAT (' ',T10,I4,T20,F13.6,T40,F13.6)                           00003630
 1013 FORMAT ('0',/////' ','FOURIER VALUE OF CIRCUMFERENTIAL ROTATION OF00003640
     1 10TH CIRCUMFERENCE'//' ',T10,'DEGREE',T20,'PHITRO',T40,'DEGREE',T00003650
     250,'PHITRO',T70,'DEGREE',T80,'PHITRO',T100,'DEGREE',T110,'PHITRO')00003660
 1014 FORMAT (' ',T10,I4,T20,F13.6,T40,I4,T50,F13.6,T70,I4,T80,F13.6,T1000003670
     10,I4,T110,F13.6)                                                  00003680
 1015 FORMAT (13I3)                                                     00003690
 1016 FORMAT (' ','THIS IS ACTUAL ARRAY OF PHIXRO FOURIER IMPERFECTION C00003700
     1OEFFS FOR USE BY SATANS-II'//)                                    00003710
 1017 FORMAT      (25(3(' ',10E13.6/),' ',5E13.6/////))                 00003720
 1018 FORMAT (' ','THIS IS ACTUAL ARRAY OF PHITRO FOURIER IMPERFECTION C00003730
     1OEFFS FOR USE BY SATANS-II'//)                                    00003740
 1019 FORMAT (175(5E14.6/))                                             00003750
 1020 FORMAT (' ','FOR THESE MODES,  ',13I3///)                         00003760
 9999 STOP                                                              00003770
      END                                                               00003780
```

```
      SUBROUTINE FORIER (LP,Y,M,AHO,SYS,A,B,E)                          00003790
C                                                                       00003800
C  GIVEN A SET OF LP DATA POINTS Y(Q), EQUALLY SPACED, FORIER WILL DETER00003810
C  MINE LP CONSTANTS SO THAT EITHER EQUATION B.1 OR B.2 BELOW IS        00003820
C  SATISFIED:                                                           00003830
C                                                                       00003840
C       IF LP BE AN ODD NUMBER, THEN N IS TAKEN AS (L-1)/2, AND;        00003850
C  B.1:                                                                 00003860
C                  P=1/¯                                               _/00003870
C   Y(N)=A(0)/2 + SUM/  A(P)COS(2PI*NP/(2N+1)) + B(P)SIN(2PI*NP/(2N+1)) /00003880
C                   N /_                                               _/00003890
C                                        WHERE N=0,1,2,...,2N           00003900
C                                                                       00003910
C       IF LP BE AN EVEN NUMBER, THEN N IS TAKEN AS L/2 AND;            00003920
C  B.2                                                                  00003930
C                N-1/¯                                                 _/00003940
C  Y(N)=A(0)/2+SUM/(A(P)COS(PI*NP/N)+B(P)SIN(PI*NP/N))+A(N)COS(PI*N))/2/00003950
C                P=1/_                                                 _/00003960
C                                        WHERE N=0,1,2,...,2N-1         00003970
C     CONTINUE                                                          00003980
C  ARGUMENTS:                                                           00003990
C     LP:  NUMBER OF DATA POINTS TO BE FITTED, AN INTEGER.  THIS        00004000
C          IS THE NUMBER OF ITEMS IN THE Y ARRAY TO BE CONSIDERED       00004010
C          BY FORIER. (DO NOT INCLUDE THE POINT AT 2PI)                 00004020
C     Y:   DATA ARRAY TO BE DECLARED IN USER'S DIMENSION STATEMENT      00004030
C          AS A REAL*4.  IT MUST CONTAIN AT LEAST LP MEANGFUL           00004040
C          ITEMS.  Y DATA ARE ALWAYS TAKEN TO BE EQUALLY SPACED.        00004050
C     M:   HIGHEST HARMONIC TO BE CONSIDERED IN MAKING THE FIT, AN      00004060
C          INTEGER.  IF M<=0 OR M>N, THEN THE EFFECTIVE M USED BY       00004070
C          FORIER WILL BE TAKEN AS EQUAL TO N.                          00004080
C     AHO: THE RESULT A(0)/2.                                           00004090
C     SYS: THE SUM OF Y**2.                                             00004100
C     A:   AN ARRAY DECLARED IN USER'S DIMENSION STATEMENT.  A(1)       00004110
C          CONTAINS THE FOURIER COEFFICIENT A1, A(2) CONTAINS A2, ETC.  00004120
C          NOTE: A(N) CONTAINS A(N/2) WHEN LP IS AN EVEN NUMBER.        00004130
C          NOTE: A MUST BE DIMENSIONED AT LEAST N IN SIZE.              00004140
C     B:   SIMILAR TO THE A ARRAY.  CONTAINS THE SINE COEFFICIENTS.     00004150
C          NOTE: WHEN LP IS EVEN, B(N) IS SET TO 0.0.                   00004160
C     E:   AN ARRAY DECLARED IN USER'S DIMENSION STATEMENT.  E(I)       00004170
C          CONTAINS THE SUM OF THE SQUARES OF THE ERRORS IN THE FIT     00004180
C          WHEN TERMS ONLY UP TO 'I' HARMONICS ARE USED.  IF N HARMONICS00004190
C          ARE USED, E(N) IS ZERO BY DEFINITION BUT IS CALCULATED       00004200
C          ANALYTICALLY TO PROVIDE SOME CONTROL ON ACCURACY OF CALC.    00004210
C                                                                       00004220
      DIMENSION Y(1),A(1),B(1),E(1)                                     00004230
      L=LP                                                              00004240
```

```
      LM1=L-1                                                          00004250
      IL=2                                                             00004260
      N=L/2                                                            00004270
      IF(2*N-L)12,10,11                                                00004280
   11 STOP                                                             00004290
   12 IL=1                                                             00004300
   10 IF(M)15,15,14                                                    00004310
   14 ME=M                                                             00004320
      IF(M-N)16,16,15                                                  00004330
   15 ME=N                                                             00004340
   16 SUMYS=0.0                                                        00004350
      DO 13 I=1,L                                                      00004360
   13 SUMYS=SUMYS+Y(I)**2                                              00004370
      SYS=SUMYS                                                        00004380
      C=1.0                                                            00004390
      Q=2.0                                                            00004400
      S=0.0                                                            00004410
      FL=L                                                             00004420
      FN=FL/2.0                                                        00004430
      CON1=6.2831853/FL                                                00004440
      C1=COS(CON1)                                                     00004450
      S1=SIN(CON1)                                                     00004460
      CON2=2.0/FL                                                      00004470
      IM=1                                                             00004480
      DO 28 J=1,N                                                      00004490
      IP=J-1                                                           00004500
  255 U2=0.0                                                           00004510
      U1=0.0                                                           00004520
      LN=L                                                             00004530
      DO 23 K=1,LM1                                                    00004540
      U0=Y(LN)+Q*U1-U2                                                 00004550
      U2=U1                                                            00004560
      U1=U0                                                            00004570
   23 LN=LN-1                                                          00004580
   25 AAA=CON2*(Y(1)+C*U1-U2)                                          00004590
      IF(IP)11,250,251                                                 00004600
  250 AH0=AAA/2.0                                                      00004610
      SUMC=AH0*AAA                                                     00004620
      GO TO 26                                                         00004630
  251 GO TO (252,252,253),IM                                           00004640
  252 A(IP)=AAA                                                        00004650
      B(IP)=CON2*S*U1                                                  00004660
      SUMC=SUMC+A(IP)**2+B(IP)**2                                      00004670
  254 E(IP)=SUMYS-FN*SUMC                                              00004680
   26 IF(IP-ME)27,33,11                                                00004690
   27 Q=C1*C-S1*S                                                      00004700
      S=C1*S+S1*C                                                      00004710
      C=Q                                                              00004720
```

```
 28 Q=C+Q                                                        00004730
    IM=IM+IL                                                     00004740
    IP=N                                                         00004750
    GO TO 255                                                    00004760
253 A(N)=AAA/2.0                                                 00004770
    B(N)=0.0                                                     00004780
    SUMC=SUMC+A(N)*AAA                                           00004790
    GO TO 254                                                    00004800
 33 RETURN                                                       00004810
    END                                                          00004820
```

LIST OF REFERENCES

1. Bushnell, D., Computer Analysis of Shell Structures, paper 69-WA/PVP-13 presented at the ASME Winter Annual Meeting, Los Angeles, California, November 1969.

2. Air Force Flight Dynamics Laboratory Report AFFDL-TR-71-54, An Assessment of Current Capability for Computer Analysis of Shell Structures, by R. F. Hartung, April 1971.

3. National Aeronautics and Space Administration Report NASA-CR-909, A Geometrically Nonlinear Analysis of Arbitrarily Loaded Shells of Revolution, by R. E. Ball, January 1968.

4. National Aeronautics and Space Administration Report NASA-CR-1987, A Computer Program for the Geometric Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution, Theory and User's Manual, by R. E. Ball, April 1972.

5. Dantone, J. J., A Computer Study of the Buckling of an Imperfect Cylindrical Shell, M.S. Thesis, Department of Aeronautics, Naval Postgraduate School, Monterey, California, September 1971.

6. National Aeronautics and Space Administration Report NASA-CR-1163, Experimental Investigation of the Effect of General Imperfections on the Buckling of Cylindrical Shells, by J. Arbocz and C. D. Babcock, Jr., September 1968.

7. Sanders, J. Jr., "Nonlinear Theories for Thin Shells," Quarterly Journal of Applied Mathematics, v. 21, p. 21-36, 1963.

8. National Advisory Committee on Aeronautics Report 1010, A Recurrance Matrix Solution for the Dynamic Response of Aircraft in Gusts, by J. C. Houbolt, 1951.

9. Mathematics Centrum, Amsterdam, Holland Report MR19, A Matrix Method for the Solution of a Second Order Difference Equation in Two Variables, by M. L. Potters, 1955.

10. Air Force Flight Dynamics Laboratory/Lockheed Missiles and Space Company Report AFFDL-TR-71-79, Computer Oriented Analysis of Shell Structures, "Design Problems of Shell Structures and the Impact of the Computer on Shell Analysis," by M. Stein, p. 14, June 1971.

11. W. R. Church Computer Center User's Manual, Appendix C, Naval Postgraduate School, Monterey, California, 1971.

12. Texas A&M University Aerospace Engineering Department Report 69-77, Nonlinear Dynamic Analysis of Shells of Revolution by Matrix Displacement Method, by J. A. Stricklin, and others, February 1970.

13. The Catholic University of America, A Note on Buckling of Spherical Caps with Initial Asymmetric Imperfections, by Robert Kao, December 1971.

14. National Aeronautics and Space Administration Report NASA-CR-161, Influence of Edge Conditions on the Stability of Axially Compressed Cylindrical Shells, by Bö O. Almroth, February 1965.

15. National Aeronautics and Space Administration Report NASA-CR-1998, A Digital Computed Study of the Buckling of Shallow Spherical Caps and Truncated Hemispheres, by W. C. Stilwell and R. E. Ball, June 1972.

16. Bushnell, D., Almroth, B. O., and Brogan, F.A., "Finite Difference Energy Method for Nonlinear Shell Analysis," Journal of Computers and Structures, v. 1, p.361-387, 1971.

17. Huang, N. C., "Unsymmetrical Buckling of Thin Shallow Shells," Journal of Applied Mechanics, v. 31, p. 447-457, September 1964.

18. Famili, J., and Archer, R. R., "Finite Asymmetric Deformation of Shallow Spherical Shells," Journal of the American Institute of Aeronautics and Astronautics, v. 3, p. 506-510, March 1965.

19. American Institute of Aeronautics and Astronautics Paper No. 68-292, Nonlinear Deflection of Asymmetrically Loaded Shells of Revolution, by H. G. Schaeffer and R. E. Ball, April 1968.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Robert E. Ball Department of Aeronautics Naval Postgraduate School Monterey, California 93940	2
4.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	LCDR Bruce A. Ryan, USN 148 Lincoln Place Waldwick, New Jersey 07463	1
6.	Professor Johann Arbocz Firestone Flight Sciences Laboratory California Institute of Technology Pasadena, California 91109	1
7.	Professor Charles Babcock Firestone Flight Sciences Laboratory California Institute of Technology Pasadena, California 91109	1
8.	Dr. Richard F. Hartung, Manager Structural Mechanics Laboratory Lockheed - Palo Alto Research Laboratories 3251 Hanover Street Palo Alto, California 94304	1
9.	Dr. Bö O. Almroth Structural Mechanics Laboratory Lockheed - Palo Alto Research Laboratories 3251 Hanover Street Palo Alto, California 94304	1
10.	Dr. Frank A. Brogan Structural Mechanics Laboratory Lockheed - Palo Alto Research Laboratories 3251 Hanover Street Palo Alto, California 94304	1

11. Dr. David Bushnell 1
Structural Mechanics Laboratory
Lockheed - Palo Alto Research Laboratories
3251 Hanover Street
Palo Alto, California 94304

12. Professor Lars Åke Samuelson 1
The Aeronautical Research Institute of Sweden
P. O. Box 11021
S-161 11 Bromma 11
SWEDEN

13. Mr. Richard Citerly 1
Anamet Laboratories
P. O. Box 831
San Carlos, California 94070

14. Dr. Robert Fulton 1
Structural Mechanics Branch
NASA - Langley Research Center
Hampton, Virginia 23365

15. Mr. John A. O'Malley 1
Code 45
Naval Weapons Center
China Lake, California 93555

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE A Digital Computer Study of the Buckling of Actual Imperfect ylinders - A Modification of the Computer Program SATANS - Theory and ser's Manual for SATANS-I and SATANS-II.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
hesis for the degree of Aeronautical Engineer, December 1972.

5. AUTHOR(S) (First name, middle initial, last name)
ruce A. Ryan, LCDR,USN

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
ecember 1972	268	19

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT
pproved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
ork was partially supported by ode 45, Naval Weapons Center, hina Lake, California 93555.	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT A digital computer program for the geometrically nonlinear stati nd dynamic response of arbitrarily loaded shells of revolution known as ATANS is modified, removing a requirement that the applied loads be ymmetric about a datum meridional plane. The ability to include rbitrary naturally-occurring initial geometric imperfections of any hape, location or extent is added to the program. Two specific axially oaded circular cylinders are analyzed using actual initial imperfec- ions. Detailed numerical studies are conducted to determine the uckling loads of the perfect and imperfect cylinders. Several sets of oundary conditions are considered. An analysis procedure for imperfect hells is proposed. The results indicate that a means to conduct a umerical analysis of an arbitrarily imperfect shell of revolution has een developed.

A study of the buckling behavior of the clamped shallow spherical ap modelled under various boundary conditions at the pole is included s an addendum.

D FORM 1 NOV 68 1473 (PAGE 1)
S/N 0101-807-6811

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
SHELLS						
SHELLS OF REVOLUTION						
STRUCTURAL ANALYSIS						
BUCKLING						
BUCKLING OF SHELLS						
BUCKLING OF IMPERFECT SHELLS						
IMPERFECTIONS						
IMPERFECT SHELLS						

Thesis
R94 Ryan
c.1
139074
A digital computer study of the buckling of actual imperfect cylinders - a modification of the computer program SATANS - theory and user's model for SATANS-I and SATANS-II.